
AREDN Documentation

Release latest

AREDN

Apr 17, 2024

GETTING STARTED GUIDE

1	AREDN® Overview	3
2	Selecting Radio Hardware	5
3	Downloading AREDN® Firmware	7
4	Installing AREDN® Firmware	11
5	Basic Radio Setup	27
6	Node Status Display	33
7	Mesh Status Display	39
8	Configuration Deep Dive	45
9	Reporting Problems or Issues	81
10	Networking Overview	83
11	Network Topologies	85
12	Radio Spectrum Characteristics	91
13	Channel Planning	97
14	Network Modeling	107
15	AREDN® Services Overview	113
16	Chat Programs	115
17	Email Programs	123
18	File Sharing Programs	127

19	VoIP Audio/Video Conferencing	131
20	Video Streaming and Surveillance	139
21	Network Management Tools	149
22	Computer Aided Dispatch	155
23	Other Services	159
24	Tips for Uploading Firmware	167
25	Connecting Nodes to Home Routers	171
26	Power over Ethernet (PoE)	173
27	Link Quality Manager (LQM)	175
28	Configuring a Supernode	181
29	Test Network Links with iperf3	187
30	Command Line Access to Your Node	191
31	Comparing SISO and MIMO Hardware	195
32	Settings for Radio Mobile	199
33	Tips for Aiming Directional Antennas	201
34	Use PuTTYGen to Make SSH Keys	205
35	Creating a Local Package Server	215
36	Using Cross Links	219
37	Virtual Machine Installs	223
38	Tools for Developers	227
39	Known Issues	233
40	Additional Information	235
41	Responsible Disclosure Policy	237
42	Frequencies and Channels	239
43	Acroynms List	241



Release
latest

This documentation set consists of several sections which are shown in the navigation list.

- The **Getting Started Guide** walks through the process of configuring an AREDN® radio node to be part of a mesh network.
- The **Network Design Guide** provides background information and tips for planning and deploying a robust mesh network.
- The **Applications and Services Guide** discusses the types of programs or services that can be used across a mesh network.
- The **How-to Guides** provide tips and techniques for various tasks.
- Finally, the **Appendix** contains supplementary information.

If you wish to locate specific topics within the documentation, you can type keywords into the *Search docs* field to display a list of items which match your search.

If you would like to see the documentation for a specific AREDN® release, click on the **Read the Docs** label at the bottom of the navigation bar. This label shows the version you are currently viewing, but clicking the label bar opens a panel with several other options. Here you may choose to view another version of the documentation, and you can also download the entire documentation set in any of several formats (*PDF, ePub, HTML*) for offline use.

Note: AREDN® is a registered trademark of *Amateur Radio Emergency Data Network, Inc.* and may not be used without permission.

AREDN® OVERVIEW

The AREDN® acronym stands for “Amateur Radio Emergency Data Network” and it provides a way for *Amateur Radio* operators to create high-speed ad hoc *Data Networks* for use in *Emergency* and service-oriented communications.

For many years amateur radio operators and their served agencies have relied on voice transmissions for emergency or event communications. A typical message-passing scenario involved conveying the message to a radio operator who would write or type it onto a standard ICS-213 form. The message would then be relayed by radio to another operator who would write or type it on another ICS-213 form at the receiving end. The form would typically be hand-delivered to the recipient who would read and sign the form. Any acknowledgement or reply would then be handled through the same process from the receiving end back to the originator.

This tried-and-true scenario has worked well, and it continues to work for handling much emergency and event traffic. Today, however, digital transmission is more commonly used instead of traditional methods and procedures. The hardcopy ICS-213 form is giving way to the Winlink electronic form, with messages being passed using digital technologies such as AX.25 packet, HF Pactor, Fldigi, and others.

Our Mission

The primary goal of the AREDN® project is to empower licensed amateur radio operators to quickly and easily deploy high-speed data networks when and where they are needed.

In today’s high-tech society people have become accustomed to different ways of handling their communication needs. The preferred methods involve short messaging and keyboard-to-keyboard communication, along with audio-video communication using Voice over IP (VoIP) and streaming technologies.

The amateur radio community is able to meet these high-bandwidth digital communication requirements by using FCC Part 97 amateur radio frequency bands to send digital data between devices which are linked with each other to form a self-healing, fault-tolerant data network. Some have described this as an amateur radio version of the Internet. Although it is not intended for connecting people to **the Internet**, an AREDN® mesh network will provide typical Internet or intranet-type

applications to people who need to communicate across a wide area during an emergency or community event.

An AREDN® network is able to serve as the transport mechanism for the preferred applications people rely upon to communicate with each other in the normal course of their business and social interactions, including email, chat, phone service, document sharing, video conferencing, and many other useful programs. Depending on the characteristics of the AREDN® implementation, this digital data network can operate at near-Internet speeds with many miles between network nodes.

A foundational design goal of the AREDN® project is to minimize the technical expertise that is normally required to configure a robust radio network. Devices running AREDN® firmware are in many ways self-configuring so that users without a background in IP networking can easily build or connect to a local RF network. As mentioned in a recent [Amateur Radio Digital Communications \(ARDC\)](#) annual report, “AREDN® software allows volunteers to set up a node with minimal expertise and effort, and because the software configures the network automatically, advanced network technology is not needed.”

This facilitates the primary goal of the AREDN® project, which is **to empower licensed amateur radio operators to quickly and easily deploy high-speed data networks when and where they are needed, as a service both to the hobby and the community.** This is especially important in cases when traditional “utility” services (electricity, phone lines, or Internet services) become unavailable. In those cases an off-grid amateur radio emergency data network may be a lifeline for communities impacted by a local disaster.

[Link: AREDN Webpage](#)

SELECTING RADIO HARDWARE

The amateur radio community has recognized the benefits of using inexpensive commercial WISP (Wireless Internet Service Provider) radios to create AREDN® networks. Each of these devices come with the vendor's firmware pre-installed, but by following a few simple steps this firmware can be replaced with an AREDN® firmware image.

Several open source software projects have been adapted and enhanced to create the AREDN® firmware, including [OpenWRT \(Open Wireless Router\)](#) and [OLSR \(Optimized Link State Routing protocol\)](#).

The AREDN® team builds specific firmware images tailored to each type of radio, and the current list of supported devices is found on the AREDN® website. For a complete list of all supported hardware, including both *Stable Release* and *Nightly Build* firmware, refer to the [Supported Devices](#) list.

When selecting a device for your AREDN® hardware there are several things to consider in your decision.

- Radios should be purchased for the specific frequency band on which they will operate. Currently AREDN® supports devices which operate in several bands. Check the [frequency and channel chart](#) on the AREDN® website for the latest information.
- Many devices have an integrated dual-polarity MIMO (Multiple Input-Multiple Output) antenna which helps to leverage multipath propagation. AREDN® has always supported and recommended using MIMO hardware, since these devices typically outperform single chain radios when used as mesh nodes.
- Radios can be purchased separately from the antenna, so it is possible to have more than one antenna option for a radio in order to optimize AREDN® nodes for varying deployment conditions.
- Costs of devices range from \$25 to several hundred dollars for a complete node/antenna system, so there are many options even for the budget-conscious operator.
- Some older or lower cost devices have a limited amount of onboard memory, but firmware images continue to grow in size and functionality. Consider purchasing a device with more memory over one with less memory.

- Check the maximum power output of the device, since some devices have lower power capabilities.

One of the best sources of detailed hardware information is a manufacturer's datasheet, usually available for download from the manufacturer's website. Currently AREDN® supports dozens of device models from manufacturers including GL-iNet, Mikrotik, TP-LINK, and Ubiquiti Networks.

If you are just getting started with AREDN® you can easily begin with one of the low-cost devices that comes with an integrated antenna and a PoE (Power over Ethernet) unit. If you are expanding your AREDN® network with more sophisticated equipment, you may choose a standalone radio attached to a high-gain antenna.

Note: See the **Network Design Guide** for more information about constructing robust mesh networks.

[Link: AREDN Webpage](#)

DOWNLOADING AREDN® FIRMWARE

3.1 Types of Firmware

Stable Release firmware has been tested and shown work on the devices that were supported at the time of the release. This firmware is considered to be stable and suitable for production devices deployed in the field. Stable Release firmware is identified by numbers such as 3.23.4.0. In this example 23.4 indicates the year (2023) and month (April) of the Stable Release.

Nightly Build firmware contains the latest bug fixes, features, and support for new devices. It allows the wider mesh community to test new code before it is included in a Stable Release. The Nightly Build is considered more experimental or cutting-edge and may not be suitable for production nodes. However, it might make sense to install the Nightly Build if you are having a specific issue that has been addressed in newly developed code or if you are loading AREDN® firmware onto a device that is newly supported. The Nightly Build filename shows the build date and the software commit identifier for that specific firmware build.

3.2 Choosing Firmware to Download

The first step is to choose the AREDN® firmware image for your specific hardware. You can find the available firmware images for your device by using the [AREDN Firmware Selector \(AFS\)](#).

Download AREDN Firmware for your Device

Type the name or model of your device, then select a stable build (ie. 3.22.12.0) or the nightly "snapshot" build (ie. 2050-781425a).

mikrotik 3.23.4.0

- MikroTik hAP ac2
- MikroTik hAP ac3
- MikroTik RouterBOARD 911G-5HPnD-QRT
- MikroTik RouterBOARD 912UAG-2HPnD
- MikroTik RouterBOARD 912UAG-5HPnD
- MikroTik RouterBOARD 921GS-5HPacD-15s (mANTBox 15s)
- MikroTik RouterBOARD 921GS-5HPacD-19s (mANTBox 19s)
- MikroTik RouterBOARD 952Ui-5ac2nD (hAP ac lite)

Enter the first few characters of the hardware manufacturer in the *Model* search field (case insensitive), then click the firmware image dropdown on the right to choose the firmware release that you want to download. Next, find your device model in the search results list and click the row for your hardware.

Download AREDN Firmware for your Device

Type the name or model of your device, then select a stable build (ie. 3.22.12.0) or the nightly "snapshot" build (ie. 2050-781425a).

MikroTik RouterBOARD LHG 5HPnD (LHG 5)

3.23.4.0

About this build

Model: **MikroTik RouterBOARD LHG 5HPnD (LHG 5)**
 Platform: ath79/mikrotik
 Version: 3.23.4.0 (r11427-9ce6aa9d8d)
 Date: 2023-04-13 09:51:05
 OpenWrt 
 Info:

Download an image



Linux kernel with minimal file system that loads to RAM. Useful for first installation or recovery on some devices.

sha256sum: 25e9a63f2a1de237e360433972a3b667b6ae487c2a82bdd898e77f5374c59cd4



Use a Sysupgrade image to update a router that already runs AREDN. The image can be used with the AREDN web interface.

sha256sum: 7337492e45c43865ba2db83196f8859707eddf282b0d9b55482002fb9f2a9122

There are usually two types of firmware images shown for each device: one for the first-time replacement of the manufacturer's firmware, and the other for upgrades of nodes that are already running AREDN® firmware.

TP-LINK or Ubiquiti

If you are loading firmware on TP-LINK or Ubiquiti devices for the first time you must download the *FACTORY* firmware. Otherwise download the *SYSUPGRADE* firmware image.

Mikrotik

If you are loading firmware on Mikrotik devices for the first time you must download **both** the *KERNEL* and *SYSUPGRADE* images. Otherwise download only the *SYSUPGRADE* firmware image.

GL.iNET

For GL.iNet devices you will only see the *SYSUPGRADE* image for both first-time installs or firmware upgrades.

Click the appropriate button to download the image file to your local computer. Make a note of the download location on your computer, since you will use the downloaded image(s) to install the

AREDN® firmware on your device.

Features Inherited from OpenWRT for New Architectures

The latest AREDN® firmware contains features which are inherited from the newest OpenWRT upstream releases. The [OpenWRT *Release Notes*](#) describe these new features. One important change is the inclusion of new *target* architectures for the firmware. The legacy “ar71xx” target has been retired and is replaced by the “ath79” and “ipq40xx” targets.

All supported devices have been migrated to the new targets. **You should select the latest recommended target image based on the type of hardware on which it will be installed.** Refer to the latest [Supported Devices](#) in order to ensure you have the correct firmware image for your specific device.

Nightly Build Direct Download

To download the *Nightly Build* directly, navigate to <http://downloads.arednmesh.org/snapshots/targets/>. Nightly Build filenames are prefixed with the firmware build date and a unique software commit identifier. As explained above, select the correct target and sub-target for the device you will be flashing. To return your device to the current stable release, download the correct *Stable Release* firmware and reflash your device.

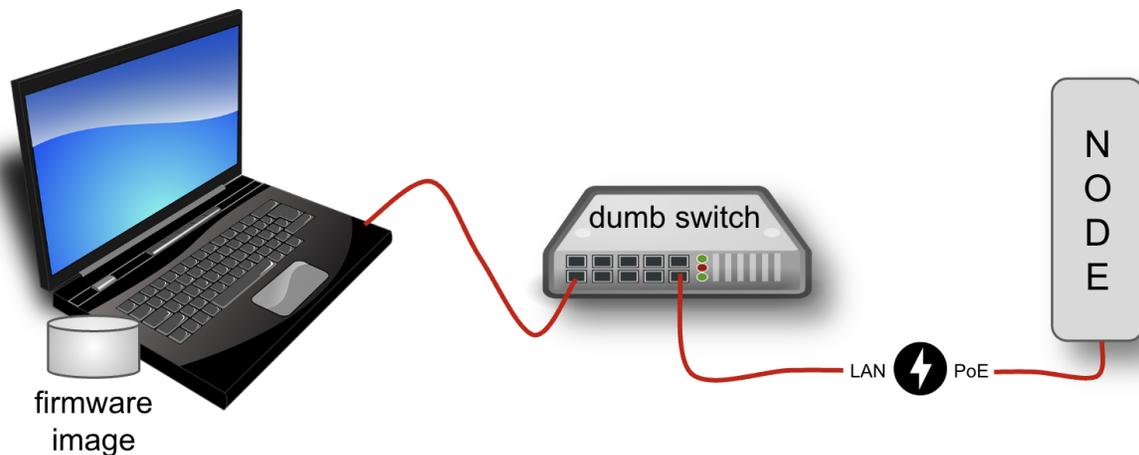
Be aware that when a new nightly build becomes available, any older builds automatically become obsolete. If you want to install add-on packages for nodes running a nightly build, understand that specific packages will not be available for an *older* build if a *newer* build has superseded it. Be sure to upgrade to the current nightly build before installing packages.

[Link: AREDN Webpage](#)

INSTALLING AREDN® FIRMWARE

There are two cases for installing AREDN® firmware:

1. If you already have an existing version of AREDN® running on your device, then you can use your computer's web browser and navigate to **Setup > Administration > Firmware Update** to install your new firmware. This process will be explained in more detail in the **Configuration Deep Dive** section of this guide. Also, see *Firmware Upgrade Tips* in the **How-to Guides** section for additional information.
2. If you are installing AREDN® firmware on a device for the first time, each hardware platform may require a unique procedure.



The diagram above shows that your computer with the downloaded firmware image must be connected to the node using Ethernet cables in order to install the AREDN® image. It is highly recommended that you connect the computer and node through a simple (dumb) Ethernet switch so that the switch can maintain the computer's network link even when the node is rebooting. Do *not* use a network router for this purpose – only a dumb switch. This is not a requirement for the sake of the radio, but may be useful for your computer to maintain its Ethernet interface link.

Different radio hardware will require different methods for installing the AREDN® firmware. For **Ubiquiti** 802.11n devices, your computer's *TFTP client* will connect to the node's *TFTP server* in order to upload the firmware image. For Ubiquiti 802.11ac devices you will follow a separate

procedure explained below. For **Mikrotik** and **TP-LINK** devices, your computer will run a **PXE server** and the node's remote boot *client* will download the boot image from your computer. For **GL-iNet** devices, your computer's web browser will connect to the node's web server to upload the firmware image. Refer to the specific procedures below for your node hardware.

In the *Firmware Tips* section of the **How-To Guide** you will find assistance if you experience an issue uploading firmware to your device. The **How-To Guide** also contains a *Virtual Machine Installs* section for help installing x86_64 firmware images on a VM for a virtualized node.

4.1 Preparing Your Computer

Setting a Static IP Address on your Computer

For all of the device models discussed below you will be asked to set a static IP address on your computer as part of the install process. Various computer operating systems have different ways of accomplishing this, and there is a wealth of information in computer manuals, publications, and online resources to walk you through the steps for your specific computer.

As mentioned above, AREDN® recommends that you connect your computer to the node through an intermediary network switch. This allows your computer to activate its Ethernet interface with the static IP address even when the node is not powered on. Since node hardware needs to be powered on/off or rebooted during the install process, the network switch will keep your computer's network interface active on its static IP address.

If you choose not to use an intermediary network switch, then you will be responsible for making sure your computer maintains an active interface with the static IP address. You may need to power on the node temporarily in order for your computer to bring up its interface, but then immediately power off the node in order to follow the installation instructions for your model. Having an intermediary network switch eliminates these headaches.

Depending on your device model you may need to have various command line tools available on your computer. The required tools are native to both Linux and MacOS computers. For Windows computers you may need to enable specific features or install appropriate programs.

Ubiquiti 802.11n Installs

Your computer should have **TFTP client** software available. If you have a Windows computer, use a web search engine to find information for your specific operating system (for example search "tftp client for windows 10"). There is a wealth of information available online for configuring your Windows computer with a TFTP client program.

- [Example 1](#)
- [Example 2](#)

Ubiquiti 802.11ac Installs

Your computer should have **ssh** and **scp** software available. *Ssh* and *scp* are native to both Linux and MacOS. The OpenSSH package (which contains both commands) can be enabled

on Windows computers. Use a web search engine to find information for your specific operating system (for example search “openssh for windows 10”). Here are some examples for enabling OpenSSH on Windows computers:

- [Example for Windows 10](#)
- [Example for Windows 11](#)
- [Example for Windows 7 & 8](#)

On Windows computers you may also use programs such as [PuTTY](#) and [WinSCP](#) to connect to your device.

Mikrotik and TP-LINK Installs

These devices are programmed to download a boot image from an external source. Your computer can run a [PXE server](#) which can give the node an IP address via [DHCP](#) as well as providing the firmware image via [TFTP](#).

If you have a Windows computer you will need to install and configure a [PXE server](#). The examples below use *Tiny PXE* which can be downloaded from erwan.labalec.fr. There may be other alternative Windows programs that accomplish the same goal, such as [ERPXE](#) or [Serva](#). For TP-LINK devices you may be able to run a simple TFTP server such as [Tftpd64](#) as explained in the TP-LINK section below.

4.2 Firmware First Install Checklists

The recommended method for installing AREDN® firmware is to download and follow the appropriate *Install Checklist* below which matches your device hardware. Additional descriptions are also provided in the sections that follow.

[GL.iNet First Install Checklist \(PDF\)](#)

[Mikrotik First Install Checklist \(PDF\)](#)

[TP-LINK First Install Checklist \(PDF\)](#)

[Ubiquiti N First Install Checklist \(PDF\)](#)

4.3 Ubiquiti 802.11n First Install Process

Download the *Install Checklist* for Ubiquiti 802.11n devices. These devices have a built-in [TFTP server](#) to which you can upload the AREDN® *factory* image. Your computer must have [TFTP client](#) software available. For more information, see the **Preparing Your Computer** section above.

Different TFTP client programs may have different command line options or flags that must be used, so be sure to study the command syntax for your TFTP client software. The example shown below may not include the specific options required by your client program.

Download the appropriate *factory* file for your device by following the instructions in the **Downloading AREDN Firmware** section of this documentation.

1. Set your computer's Ethernet network adapter to a static IP address that is a member of the correct subnet for your device. Check the documentation for your specific hardware to determine the correct network number. As in the example below, most Ubiquiti devices have a default IP address of 192.168.1.20, so you can give your computer a static IP on the 192.168.1.x network with a netmask of 255.255.255.0. For example, set your Ethernet adapter to a static IP address of 192.168.1.10.

You can choose any number for the fourth octet, as long as it is not the same as the IP address of the node. Of course you must also avoid using 192.168.1.0 and 192.168.1.255, which are reserved addresses that identify the network itself and the broadcast address for that network. Other devices may have different default IP addresses or subnets, so select a static IP for your computer which puts it on the same subnet but does not conflict with the default IP of the device.

2. Connect an Ethernet cable from your computer to the dumb switch, and another cable from the LAN port of the PoE adapter to the switch.
3. Put the Ubiquiti device into TFTP mode by holding the reset button while plugging your node's Ethernet cable into the *POE* port on the PoE adapter. Continue holding the device's reset button for approximately 30 to 45 seconds until you see the LEDs on the node alternating in a 1-3, 2-4, 1-3, 2-4 pattern, then release the reset button.
4. Open a command window on your computer and execute a file transfer command to send the AREDN® firmware to your device. Target the default IP address of your Ubiquiti node, such as 192.168.1.20 (or 192.168.1.1 for AirRouters). The following is one example of TFTP commands that transfer the firmware image to a node:

```
[Linux/Mac]
> tftp 192.168.1.20
> bin [Transfer in "binary" mode]
> trace on [Show the transfer in progress]
> put <full path to the firmware file>
  [For example, put /tmp/aredn-<release>-factory.bin]
-----
[Windows with command on a single line]
> tftp.exe -i 192.168.1.20 put C:\temp\aredn-<release>-factory.
->bin
```

The TFTP client should indicate that data is being transferred and eventually completes.

5. The node will now automatically reboot with the new AREDN® firmware image.

4.4 Ubiquiti 802.11ac First Install Process

Note: The install process for these devices requires detailed steps that are best followed using the procedure below, so no separate *Install Checklist* is provided for Ubiquiti 802.11ac devices.

Prerequisites

The installing computer must be capable of connecting to the command line of the target device. This will require that the computer support both the *ssh* and *scp* protocols. *SSH* and *scp* are native to both Linux and MacOS. The OpenSSH package (which contains both commands) can be enabled on Windows computers. For more information, see the **Preparing Your Computer** section above.

Step 1: Preparing the device

Before you install AREDN® firmware on a Ubiquiti 802.11ac device, you must first make sure it is running a specific version of the standard Ubiquiti AirOS software. This procedure will not work if the device is running any other version. Fortunately you can upgrade or downgrade the standard Ubiquiti software.

As described in the first paragraphs of this document, it is best to connect your computer to the device using a simple Ethernet switch so that your computer's network interface remains unaffected by reboots on the radio. The IP address for a new Ubiquiti device is 192.168.1.20. Set the IP address of your computer to 192.168.1.10 and, when the device is powered up, enter 192.168.1.20 in a web browser. For a brand new device you'll be asked to select your country and agree to the EULA. Then click *Continue*. Next you will be prompted to create a user account and password on the radio. You can enter the username `admin` and the password `admin!23` (for example) and then click *Save*. Make a note of this username and password because you will use it in the following steps.

You should now see the main Dashboard view in AirOS. On the left, click the *Gear* icon. This will take you to the System page. At the top of this page you will find the radio's current firmware version. For example, it might read `FIRMWARE VERSION XC.V8.7.1`. If the firmware version shows either **XC.V8.7.0** or **WA.V8.7.0** then you have the correct AirOS software and can move on to **Step 2**.

But if you see any version other than 8.7.0 you must upload new firmware to the device. You will need to download the correct firmware to your installing computer. The firmware can be found here:

- WA: <https://dl.ubnt.com/firmwares/XC-fw/v8.7.0/WA.v8.7.0.42152.200203.1256.bin>
- XC: <https://dl.ubnt.com/firmwares/XC-fw/v8.7.0/XC.v8.7.0.42152.200203.1256.bin>

Select the firmware appropriate for your device. If the radio's current firmware starts with *WA* download that version. If it starts *XC* download that version.

On the top right of the System page you will see "UPLOAD FIRMWARE" and **UPLOAD** in

blue. Clicking the blue **UPLOAD** text will open a dialog and let you select the **8.7.0** firmware you downloaded to your computer. Now that firmware will be uploaded to the device. Once completed a dialog in the top right will be displayed allowing you to either **UPDATE** or **DISCARD** the newly uploaded firmware. Click *UPDATE*. The upgrade process will now start. Do **not** unplug the device until this step is completed.

Once the upgrade has been completed, the device will return you to the login page. Log in using the username and password you created earlier (`admin / admin!23`). Once again you will see the System page and if everything has been successful, the firmware version will now read either `WA.V8.7.0` or `XC.V8.7.0` and you can move to **Step 2**.

Attention: The upgrade can fail on newer hardware which requires **8.7.4** firmware. This problem has only been observed and tested on newer LiteBeam 5AC devices. For these devices, follow the same firmware downgrade procedure but use the following firmware instead:

- **WA:** <https://dl.ubnt.com/firmwares/XC-fw/v8.7.4/WA.v8.7.4.45112.210415.1103.bin>

The rest of the process remains unchanged, so once the downgrade is successful you can move to **Step 2**.

Step 2: Copy the AREDN® firmware to the device

Before you can install AREDN® firmware on the device, you first need to put the AREDN® image in the device's `/tmp` directory. Note that each 802.11ac model will have a *different* AREDN® image name, as opposed to past releases where one AREDN® image supported multiple models. Be sure to download the correct firmware image from the AREDN® download site. On your computer, open a terminal session (“CMD” in windows). Copy the firmware to the device using the `scp` command with the username and password you created in **Step 1**. The example command below shows the placeholder `<aredn-image-factory.bin>` for the firmware filename, but be sure to replace this with the actual filename of the firmware you are installing.

```
scp <aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

If you see the error “Unable to negotiate” it means that the SCP program you are using on your computer does not support the default security key type being used on the device. You should refer to the documentation for that SCP program to resolve the issue. You can try the following:

```
scp -oHostKeyAlgorithms+=ssh-rsa -oPubkeyAcceptedAlgorithms+=ssh-rsa  
↪<aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

If you see an error “sftp-server: not found” you can try the following:

```
scp -O -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-
↪rsa <aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

If you see an error “Remote host identification has changed” you can try the following:

```
scp -O -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-
↪rsa -oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no
↪<aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

Once this is successful, the AREDN® firmware will be in /tmp on the device waiting to be installed.

Step3: Install the firmware

The installation procedure requires you to **ssh** to the command line of the device. On your computer, open a terminal session (“CMD” in windows). Type or copy/paste the following command:

```
ssh admin@192.168.1.20
```

If you see the error “Unable to negotiate” please try the following:

```
ssh -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-rsa
↪admin@192.168.1.20
```

If you see an error “Remote host identification has changed” you can try the following:

```
ssh -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-rsa
↪-oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no
↪admin@192.168.1.20
```

You will be asked for the password created in **Step 1** (for example, admin!23) and once entered you will be logged into the device and shown the shell prompt.

To install the AREDN® firmware you first need to create a program to do this. Ubiquiti devices expect signed firmware but AREDN® is not signed, so we need to bypass the checking process. To do this type or copy/paste the following two commands:

```
hexdump -Cv /bin/ubntbox | sed 's/14 40 fe 27/00 00 00 00/g' |
↪hexdump -R > /tmp/fwupdate.real

chmod +x /tmp/fwupdate.real
```

These commands take the standard Ubiquiti program used for flashing new firmware and change a few bytes to create our own version with the signature checking code disabled. The first command can take a little while to complete but when successful will return you to the shell prompt.

Finally flash the AREDN® firmware by typing:

```
/tmp/fwupdate.real -m /tmp/factory.bin
```

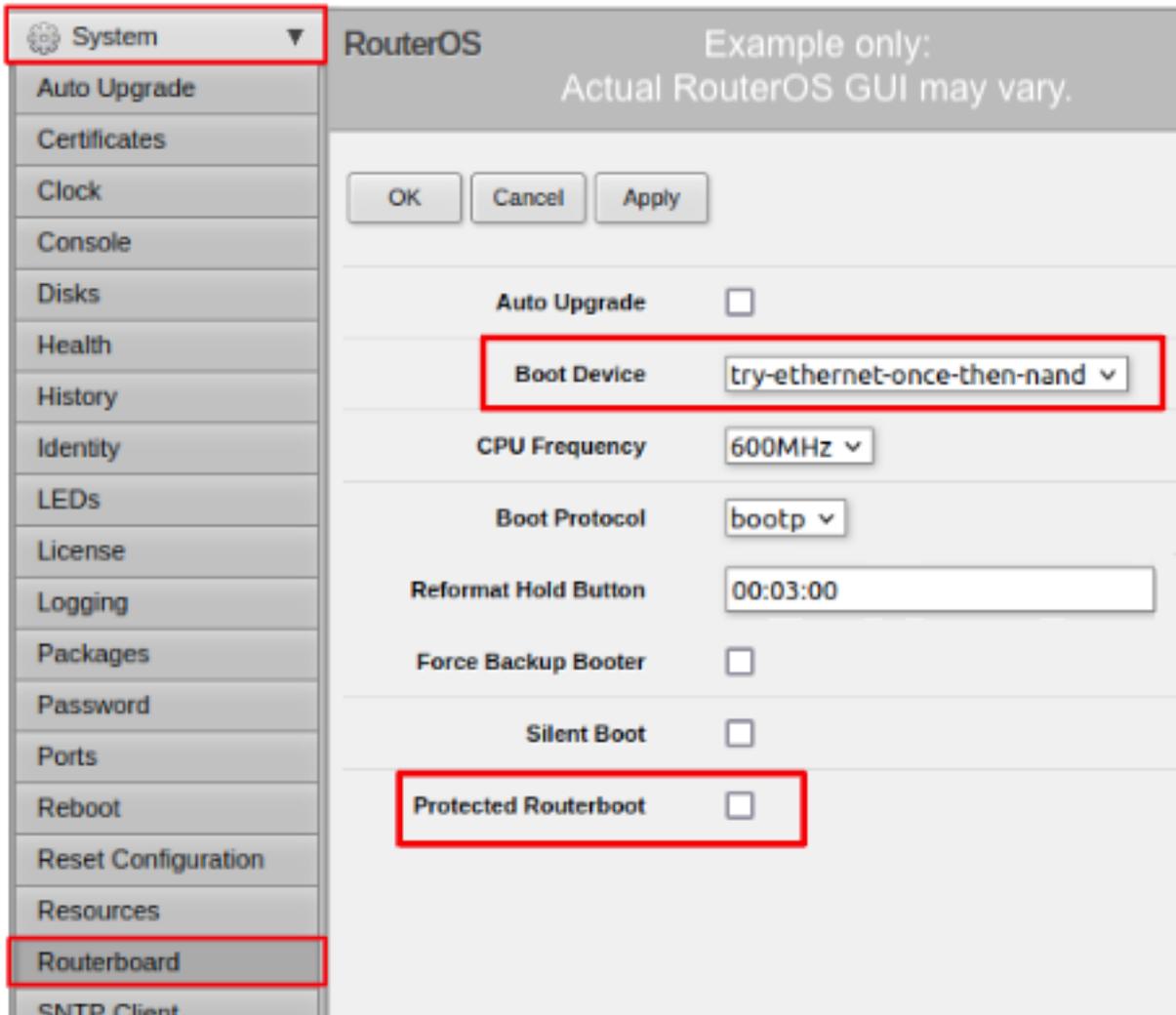
Do **not** unplug the device until the flashing process is complete and the device has rebooted. The device will install the AREDN® image, boot into it, and end up on IP address 192.168.1.1 as a normal AREDN® device. If you cannot connect to the device on its new IP address after five minutes, power cycle the device and try connecting to 192.168.1.1 again. You can then configure the device by following the steps in the **Basic Radio Setup** section of the documentation.

4.5 Mikrotik First Install Process

Download the *Install Checklist* for Mikrotik devices. These devices require a **two-part install** process: First, boot the correct Mikrotik initramfs-kernel file, and then use that temporary AREDN® Administration environment to complete the installation of the appropriate *sysupgrade* file.

Mikrotik devices have a built-in *PXE client* which allows them to download a boot image from an external source. You will need to install and configure a *PXE server* on your Windows computer. The example below uses *Tiny PXE*. For more information, see the **Preparing Your Computer** section above. For most Mikrotik devices the install steps below will work without issue.

For Mikrotik devices you will use what is called *Etherboot* mode, and there are several ways to put your device into *Etherboot* mode (depending on the version of the manufacturer's firmware it is running). The easiest way is to use the device's reset button as described in the procedure below. If for some reason this does not work, then you can try logging into the Mikrotik RouterOS and setting *System > Routerboard > Settings > Boot Device* to `try-ethernet-once-then-nand` (either through the RouterOS web interface or via command line). Next time the device boots it will try *Etherboot* once before defaulting back to regular boot mode.



Potential RouterOS Issue

If your Mikrotik device has “Protected Routerboot” enabled, then you will need to disable it before proceeding. Use the manufacturer’s instructions to connect to your device and display the RouterOS web interface or command line. Navigate to *System > Routerboard > Settings > Boot Device* to uncheck or deselect Protected Routerboot. Click the *Apply* button, then you should be able to power down the device and continue with the steps in the AREDN® firmware install checklist.

<input type="button" value="Upgrade"/> <input type="button" value="Settings"/> <input type="button" value="USB Power Reset"/>	
Routerboard	<input checked="" type="checkbox"/>
Model	RB952Ui-5ac2nD
Revision	
Serial Number	C5600DC7FAEA
Firmware Type	qca9531L
Factory Firmware	6.44
Current Firmware	7.7
Upgrade Firmware	6.45.7

You may experience an issue during installation of the *sysupgrade.bin* file on Mikrotik devices having RouterOS v7. If your Mikrotik device came with a *Current Firmware* version of v7.x you can follow the instructions on this page ([OpenWRT - downgrading RouterOS](#)) to downgrade Mikrotik RouterOS prior to flashing the AREDN® firmware. You can find the earlier firmware on the [Mikrotik Download Archive](#). Download the ARM version (routeros-arm) for devices that use the *ipq40xx* AREDN® firmware, or download the MIPSBE version (routeros-mipsbe) for other Mikrotik devices. You need to download a v6 RouterOS version that is equal or newer than the RouterOS version shown in the *Factory Firmware* field on your device (as in the example image).

Install Preparation

- Download *both* of the appropriate Mikrotik *factory* and *sysupgrade* files from the AREDN® website. Rename the *initramfs-kernel* file to *rb.elf* and keep the *sysupgrade bin* file available for later.
- Set your computer's Ethernet network adapter to a static IP address on the subnet you will be using for the new device. This can be any network number of your choice, but it is recommended that you use the 192.168.1.x subnet. Using the 192.168.1.x network on your **PXE** server will avoid changing IP addresses on your computer during the install process. AREDN® firmware uses the 192.168.1.x network once it is loaded, so using it all the way through the process will simplify things for you. For example, you can give your computer a static IP such as 192.168.1.10 with a netmask of 255.255.255.0. You can choose any number for the fourth octet, as long as it is *not* within the range of DHCP addresses you will be providing as shown below.

- Connect an Ethernet cable from your computer to the network switch as described and shown in the graphic at the top of this document, then connect another cable from the LAN port of the PoE adapter to the switch. Finally connect an Ethernet cable from the POE port to the node, but leave the device powered off for now. If you are flashing a device which uses a separate power adapter (such as a *Mikrotik hAP ac* family device), connect the last Ethernet cable from the switch to the device’s WAN port [1].

PXE Boot: *Linux Procedure*

1. Create a directory on your computer called `/tftp` and copy the `rb.elf` file there.
2. Determine your computer’s Ethernet *interface name* with `ifconfig`. It will be the interface you set to 192.168.1.10 above. You will use this interface name in the command below as the name after `-i` and you must substitute your login user name after `-u` below. Use a `dhcp-range` of IP addresses that are also on the same subnet as the computer: for example 192.168.1.100,192.168.1.200 as shown below.
3. Open a terminal window to execute the following `dnsmasq` command with escalated privileges:

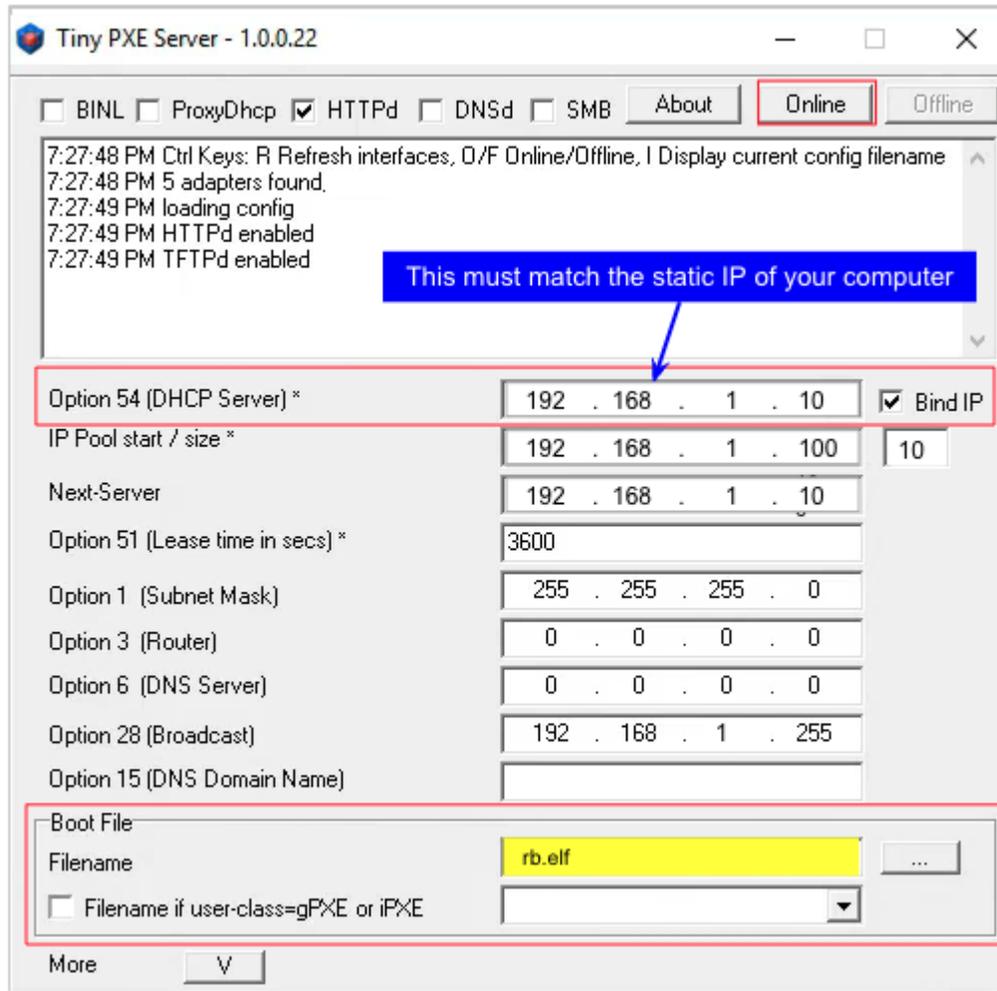
```
> sudo dnsmasq -i eth0 -u joe --log-dhcp --bootp-dynamic --dhcp-
↪range=192.168.1.100,192.168.1.200 -d -p0 -K --dhcp-boot=rb.elf --
↪enable-tftp --tftp-root=/tftp/
```

4. With the unit powered off, press and hold the reset button on the radio while powering on the device. Continue to hold the reset button until you see output information from the computer window where you ran the `dnsmasq` command, which should happen after 20-30 seconds. Release the reset button when you see the “sent” message, which indicates success, and you can now `<ctrl>-C` or end `dnsmasq`.
5. The node will now automatically reboot with the temporary AREDN® Administration image.

PXE Boot: *Windows Procedure*

Configure the PXE Server on your Windows computer. The example below uses *Tiny PXE*. For more information, see the **Preparing Your Computer** section above.

1. Navigate to the folder where you extracted the *Tiny PXE* software and edit the `config.ini` file. Directly under the `[dhcp]` tag, add the following line: `rfc951=1` then save and close the file.
2. Copy the `rb.elf` file into the `files` folder under the *Tiny PXE* server directory location.
3. Start the *Tiny PXE* server exe and select your computer’s Ethernet IP address from the dropdown list called `Option 54 [DHCP Server]`, making sure to check the `Bind IP` checkbox. Under the “Boot File” section, enter `rb.elf` into the `Filename` field, and uncheck the checkbox for “Filename if user-class = gPXE or iPXE”. Click the *Online* button at the top of the *Tiny PXE* window.



4. With the unit powered off, press and hold the reset button on the node while powering on the device. Continue holding the reset button until you see TFTPd: DoReadFile: rb.elf in the *Tiny PXE* log window.
5. Release the node's reset button and wait for the image to be transferred to the device. You are finished using *Tiny PXE* when the firmware image has been read by the node, so you can click the *Offline* button in *Tiny PXE*.
6. The node will now automatically reboot with the temporary AREDN® Administration image.

Install the *sysupgrade* Firmware Image

1. After booting the **elf** image the node will have a default IP address of 192.168.1.1. Your computer should already have a static IP address on this subnet, but if not then give your computer an IP address on this subnet.

Attention: For the *Mikrotik hAP ac* family of devices, disconnect the Ethernet cable from the WAN port (1) on the Mikrotik and insert it into one of the LAN ports (2,3,4) before you proceed.

You should be able to ping the node at 192.168.1.1. Don't proceed until you can ping the node. You may need to disconnect and reconnect your computer's network cable to ensure that your IP address has been reset. Also, you may need to clear your web browser's cache in order to remove cached pages remaining from your node's previous firmware version.

2. In a web browser, open the node's Administration page `http://192.168.1.1/cgi-bin/admin` (user = 'root', password = 'hsmm') and immediately navigate to the *Firmware Update* section. Browse to find the *sysupgrade bin* file you previously downloaded and click the *Upload* button.

As an alternative to using the node's web interface, you can manually copy the *sysupgrade bin* file to the node and run a command line program to install the firmware. This will allow you to see any error messages that may not appear when using the web interface. Note that devices running AREDN® firmware images use port 2222 for secure copy/shell access.

Execute the following commands from a Linux computer:

```
my-computer:$ scp -P 2222 <aredn-firmware-filename>.bin.
↪root@192.168.1.1:/tmp
my-computer:$ ssh -p 2222 root@192.168.1.1
~~~~~ after logging into the node with ssh ~~~~~
node:# sysupgrade -n /tmp/<aredn-firmware-filename>.bin
```

To transfer the image from a Windows computer you can use a *Secure Copy* program such as *WinSCP*. Then use a terminal program such as *PuTTY* to connect to the node via ssh or telnet in order to run the *sysupgrade* command shown as the last line above.

3. The node will now automatically reboot with the new AREDN® firmware image.

4.6 TP-LINK First Install Process

Download the *Install Checklist* for TP-LINK devices. These devices may allow you to use the manufacturer's native *PharOS* web browser interface to apply new firmware images. If available, this is the most user-friendly way to install AREDN® firmware. Navigate to the system setup menu to select and upload new firmware. Check the TP-LINK documentation for your device if you have questions about using their built-in user interface. If this process works then you will have AREDN® firmware installed on your device and you skip all of the steps described below.

If the process above does not work or if you choose not to use the *PharOS* web interface, then you can install AREDN® firmware on your device using steps similar to those described above for Mikrotik devices. TP-LINK devices are programmed to use TFTP for downloading a boot image from an external source. If you already have a PXE server on your Windows computer then you can use that. The example below uses *Tiny PXE*. It may also be possible to use a simple TFTP server instead. For more information, see the **Preparing Your Computer** section above.

Install Preparation

- Download the appropriate TP-LINK *factory* file and rename this file as `recovery.bin`
- Set your computer's Ethernet network adapter to a static IP address of 192.168.0.100.
- Connect an Ethernet cable from your computer to the network switch, and another cable from the LAN port of the PoE adapter to the switch. Finally connect an Ethernet cable from the *POE* port to the node, but leave the device powered off for now.

Linux Procedure

1. Create a directory on your computer called `/tftp` and copy the TP-LINK `recovery.bin` file there.
2. Determine your computer's Ethernet interface name with `ifconfig`. It will be the interface you set to 192.168.0.100 above. You will use this interface name in the command below as the name after `-i` and you must substitute your login user name after `-u` below. Use a `dhcp-range` of IP addresses that are also on the same subnet as the computer: for example 192.168.0.110,192.168.0.120 as shown below.
3. Open a terminal window to execute the following `dnsmasq` command with escalated privileges:

```
> sudo dnsmasq -i eth0 -u joe --log-dhcp --bootp-dynamic --  
→dhcp-range=192.168.0.110,192.168.0.120 -d -p0 -K --dhcp-  
→boot=recovery.bin --enable-tftp --tftp-root=/tftp/
```

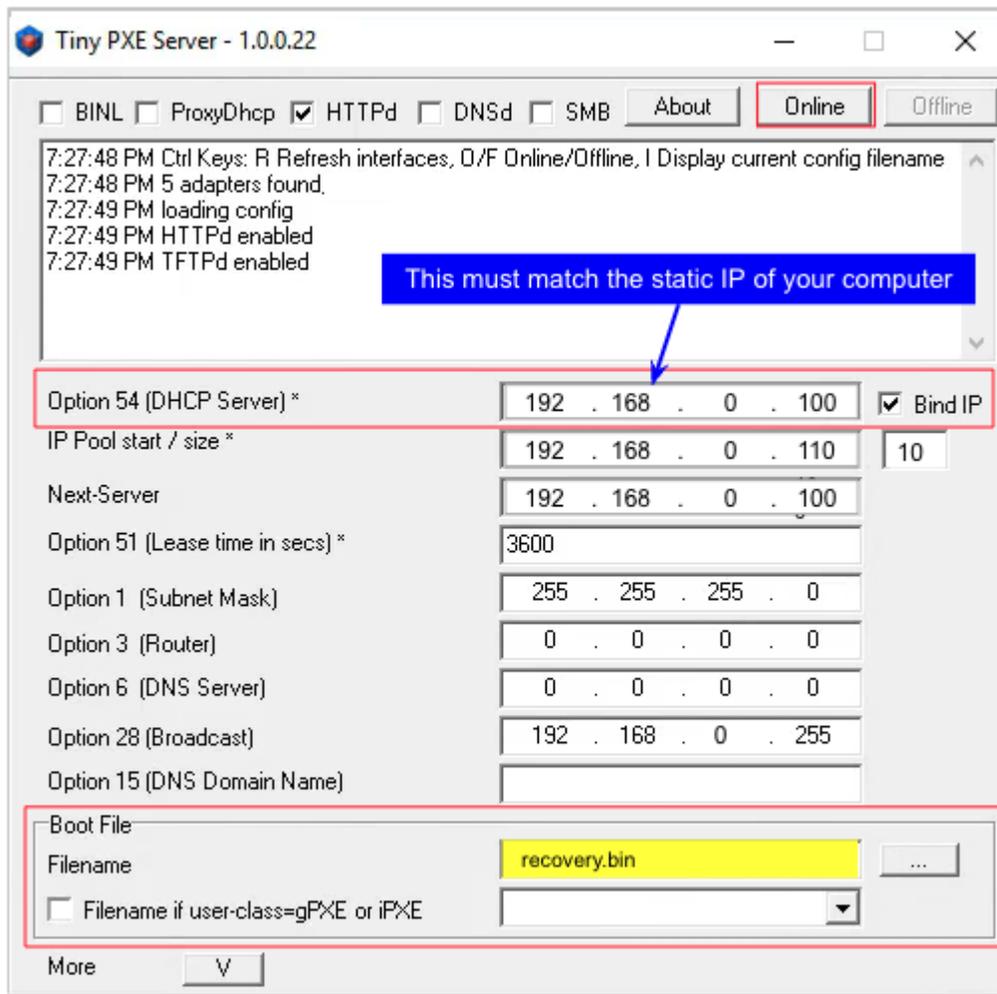
4. With the unit powered off, press and hold the reset button on the radio while powering on the device. Continue to hold the reset button until you see output information from the computer window where you ran the `dnsmasq` command, which should happen after 20-30 seconds. Release the reset button when you see the “sent” message, which indicates success, and you can now `<ctrl>-C` or end `dnsmasq`.
5. The node will now automatically reboot with the new AREDN® firmware image.

Windows Procedure

Configure the PXE or TFTP Server on your Windows computer. The example below uses *Tiny PXE*. For more information, see the **Preparing Your Computer** section above.

1. Navigate to the folder where you extracted the *Tiny PXE* software and edit the `config.ini` file. Directly under the `[dhcp]` tag, add the following line: `rfc951=1` then save and close the file.

2. Copy the `recovery.bin` firmware image into the `files` folder under the *Tiny PXE* server directory location.
3. Start the *Tiny PXE* server exe and select your computer's Ethernet IP address from the dropdown list called Option 54 [DHCP Server], making sure to check the Bind IP checkbox. Under the "Boot File" section, enter `recovery.bin` into the the *Filename* field, and uncheck the checkbox for "Filename if user-class = gPXE or iPXE". Click the *Online* button at the top of the *Tiny PXE* window.



4. With the unit powered off, press and hold the reset button on the node while powering on the device. Continue holding the reset button until you see TFTPd: DoReadFile: recovery.bin in the *Tiny PXE* log window.
5. Release the node's reset button and wait for the image to be transferred to the device. You are finished using *Tiny PXE* when the firmware image has been read by the node, so you can click the *Offline* button in *Tiny PXE*.
6. The node will now automatically reboot with the new AREDN® firmware image.

4.7 GL-iNet First Install Process

Download the *Install Checklist* for GL-iNet devices. These devices allow you to use the manufacturer's pre-installed *OpenWRT* web interface to upload and apply new firmware images. Check the GL-iNet documentation for your device if you have questions about initial configuration. Both GL-iNet and AREDN® devices provide DHCP services, so you should be able to connect your computer and automatically receive an IP address on the correct subnet. GL-iNet devices usually have a default IP address of 192.168.8.1, so if for some reason you need to give your computer a static IP address you can use that subnet.

After the GL-iNet device is first booted and configured, navigate to the **Upgrade** section and click *Local Upgrade* to select the AREDN® *sysupgrade.bin* file you downloaded for your device.

Attention: Be sure to uncheck the **Keep Settings** checkbox, since GL.iNet settings are incompatible with AREDN® firmware. Also, the AR300M16 devices may have a *boot_dev* switch, so be sure to read the [GL.iNet boot documentation](#) to select the correct boot mode.

The node will automatically reboot with the new AREDN® firmware image. If for some reason your GL-iNet device gets into an unusable state, you should be able to recover using the process documented here: [GL-iNet debrick procedure](#)

4.8 After the Firmware Install

After the node reboots, it should have a default IP address of 192.168.1.1. Make sure your computer has an IP address on the 192.168.1.x network. You should be able to ping the node at 192.168.1.1. Don't proceed until you can ping the node. You may need to disconnect and reconnect your computer's network cable to ensure that it has a connection.

Once your device is running AREDN® firmware, you can display its web interface by navigating to either `http://192.168.1.1` or `http://localnode.local.mesh`. Some computers may have DNS search paths configured that require you to use the [fully qualified domain name \(FQDN\)](#) to resolve *localnode* to the mesh node's IP address. You may need to clear your web browser's cache in order to remove any cached pages.

You can use your web browser to configure the new node with your callsign, admin password, and other settings as described in the **Basic Radio Setup** section of the documentation.

[Link: AREDN Webpage](#)

BASIC RADIO SETUP

5.1 First-Time Setup

After you have installed the AREDN® firmware and rebooted the device, the node will have a default IP address of 192.168.1.1. Make sure your computer has an IP address on the 192.168.1.x network. After connecting your computer to a LAN port on the node or the PoE unit, you should be able to ping the node at 192.168.1.1. Navigate to your node’s web interface at <http://192.168.1.1> or <http://localnode.local.mesh>. Some computers may have DNS search paths configured that require you to use the [fully qualified domain name \(FQDN\)](#) to resolve *localnode* to the mesh node’s IP address. Each node will serve its web interface on ports 80 and 8080.

The initial status page will be displayed, instructing you to configure your node by clicking the **Setup** button. This is sometimes referred to as the “NOCALL” or *firstboot* display.

NOCALL-14-144-234

Location Not Available

[Help](#)

This node is not yet configured.

Go to the setup page and set your node name and password.

Click Save Changes, even if you didn't make any changes, then the node will reboot.

This device can be configured to either permit or prohibit known encrypted traffic on its RF link. It is up to the user to decide which is appropriate based on how it will be used and the license under which it will be operated. These rules vary by country, frequency, and intended use. You are encouraged to read and understand these rules before going further.

This device is pre-configured with no restrictions as to the type of data being passed.

You will be prompted to enter the administrative login credentials. The default authentication credentials are:

Username: root

Password: hsmm

The **Basic Setup** page will be displayed, as shown below.

Node Name	<input type="text" value="AD5BC-Node2"/>	Password	<input type="password"/>
Node Description (optional)	<input type="text"/>	Verify Password	<input type="password"/>

Mesh	LAN	WAN
Enable <input checked="" type="checkbox"/>	LAN Mode: <input type="text" value="5 host Direct"/>	Protocol: <input type="text" value="DHCP"/>
Band: <input type="text" value="5GHz"/>	IP Address: 10.231.105.113	DNS 1: <input type="text" value="8.8.8.8"/>
IP Address: <input type="text" value="10.92.237.46"/>	Netmask: 255.255.255.248	DNS 2: <input type="text" value="8.8.4.4"/>
Netmask: <input type="text" value="255.0.0.0"/>	DHCP Server: <input checked="" type="checkbox"/>	
SSID: <input type="text" value="AREDN"/> -10-v3	DHCP Start: 114	
Channel: <input type="text" value="36 (5180)"/>	DHCP End: 118	
Channel Width: <input type="text" value="10 MHz"/>		
Power & Link Quality	LAN Access Point	WAN Wifi Client
Tx Power: <input type="text" value="22 dBm"/>	Enable: <input type="checkbox"/>	Enable: <input type="checkbox"/>
Max Distance: <input type="text" value="50.0"/> miles	AP band: 2GHz	SSID: <input type="text"/>
Min SNR: <input type="text" value="15"/> dB	SSID: <input type="text" value="NoCall-AREDN"/>	Password: <input type="password"/>
Min Quality: <input type="text" value="50"/> %	Channel: <input type="text" value="1"/>	
<input type="button" value="Apply"/>	Encryption: <input type="text" value="WPA2 PSK"/>	
	Password: <input type="password"/>	

Many of these settings will be described in detail in subsequent sections of this documentation. In order to get your new AREDN® node on the air for the first time, you need to enter the following items.

Node Name

Begin the node name with your callsign, followed by unique identifying information of your choice. Node names may contain up to 63 letters, numbers, and dashes, but cannot begin or end with a dash. Underscores, spaces, or any other characters are not allowed. Node names are not case sensitive, but the case will be preserved on the node status display. Amateur radio operators are required to identify all transmitting stations. The AREDN® node name is beamed automatically by the node every five minutes, so the node name must contain your call-

sign. Recommended names follow the (callsign)-(label) format, such as AD5BC-MOBILE or AD5BC-120SE. As a general rule node names should be kept as short as possible, while clearly and uniquely identifying the node.

Password

Set a new administration password for the node. Typically passwords may contain the characters a-z, A-Z, 0-9, period ., dash -, underscore _, exclamation !, and tilde ~. Avoid linux-reserved characters, including but not limited to #, \$, &, *, <, >. Enter your new password again in the *Retype Password* box to verify it is correct. You can click the *eye* icon at the right of the password fields to toggle between hidden and visible text. The first time a node is configured it will require you to change the password. Be sure to remember or record the new password so you can use it for any future administrative tasks on the node.

Node Description

This is not a required field, but it is a good place to describe the features or function of this device. Many operators use this field to list their contact information, the radio model and antenna specifications, or the tactical purpose for the node. There are no character restrictions in the field, but the maximum length allowed is 210 characters.

Mesh

The *IP Address*, *Netmask*, and *SSID* fields are automatically calculated for you based on the unique MAC (Media Access Control) address of your node. Do not change these settings. Everything under the **LAN** and **WAN** columns can be left unchanged for now.

Channel and Channel Width

Nodes communicate only with other nodes that use the same SSID, channel and channel width. You can determine the correct settings by talking with other local node operators to find out which settings are required for joining their networks.

See the **Configuration Deep Dive** section for more information about these and other settings in the *Mesh* column.

Power & Distance Settings

If you have *Link Quality Manager* disabled, you will see the *Power & Distance* settings.

- Use the dropdown list to select the maximum output power for this device. Remember that amateur operators are required to use the minimum power necessary to make contact with other stations.
- Use the slider to select the maximum distance you estimate between your node and other neighboring nodes. The default value is *zero* which tells the node to automatically determine the correct distance value to the farthest neighbor node.
- Some devices have max power levels that change depending on the channel or frequency being used, and in that case the max level may change when you save the settings. The output power will be capped at the max level supported by the hardware for that frequency.

Power & Link Quality Settings

If you have *Link Quality Manager* enabled, you will see the *Power & Link Quality* settings.

- Use the dropdown list to select the maximum output power for this device. Remember that amateur operators are required to use the minimum power necessary to make contact with other stations.
- *Max Distance* is the maximum distance between nodes at which you can expect to achieve a usable radio link. The default value is 50 miles / 80 kilometers, but you can adjust this setting if your node is only able to maintain a usable radio link with nearby nodes.
- *Min SNR* is the minimum Signal-to-Noise ratio that you require in order to reliably pass data between nodes. The default is 15 dB, but you can lower this value if you require your node to continue passing data even on links that have reduced signal characteristics.
- *Min Quality* is the minimum Link Quality required in order to reliably pass data between nodes. This is calculated as the moving average of total sent packets over total sent packets plus retransmissions. For example, if the node had to send every packet twice for it to be successfully received, the link quality would be 50%.

Once you have entered, applied, and verified that your node settings are correct, click the **Save Changes** button. Your node will record the new configuration settings and automatically reboot.

5.2 Optional Settings

Location Settings

In this section you can enter your node's latitude and longitude, as well as the grid square designator. The values should be in decimal format (for example, 30.5432 and -95.1234). The node location settings are optional, but if you have *Link Quality Manager* enabled then the location becomes important for calculating the distance between linked nodes.

Optional Settings

Latitude	<input type="text" value="33.383"/>	<input type="button" value="Find Me!"/>	<input type="button" value="Apply Location Settings"/>	<input type="button" value="Show Map"/>	<input type="button" value="Upload data to AREDN Servers"/>
Longitude	<input type="text" value="-111.505"/>	Grid Square	<input type="text" value="DM43fj"/>		
Azimuth	-	Antenna	1.5 dBi Omni		
Elevation	-	Height	<input type="text"/>		

There are several options for setting your node's location:

- If you are using a location-aware web browser, you can click the **Find Me** button to populate the latitude/longitude fields. This works well if you are viewing the *Basic Settings* page on a mobile device with built-in GPS.
- If your node has an Internet connection available, the **Show Map** and **Upload Data to AREDN Servers** buttons will become active. The **Show Map** button will display a map that allows you to click the position where your node is located or to drag an existing location

marker to a different spot on the map. Both of these actions will automatically update the latitude/longitude fields on the page.

- The **Upload Data to AREDN Servers** button will send your node information to an AREDN® server on the Internet. By submitting this information you are agreeing to allow AREDN® to publish your node location on a public mapping service and utilize the information for other purposes such as statistical analysis. No sensitive data such as passwords are sent to the AREDN® servers. If you wish to remove your node location from the public mapping service, simply clear or erase your latitude/longitude values, click *Apply Location Settings* and then *Upload Data to AREDN Servers*.
- **Antenna** information can also be entered for your node. The antenna type itself may be automatically populated based on your radio model. You may also enter the following deployment characteristics of your antenna: *Azimuth*, *Elevation* (up/down tilt), and *Height* (above ground level). Some values may not apply, such as azimuth and elevation if your node has an omnidirectional antenna.
- Click the **Apply Location Settings** button after entering new location information on this page. The new settings become active without clicking the *Save Changes* button.

Timezone and NTP Server

Here you select the timezone for your node’s system clock, and the default value is UTC. You can also enter the hostname for a [Network Time Protocol \(NTP\)](#) source if your node is connected to a network which has a network time server. In the *NTP Server* field you should enter a valid hostname for the network time source, for example `us.pool.ntp.org` or `AD5BC-ntp.local.mesh`. You may also choose how often NTP will update the node’s clock by selecting a value from the dropdown list. The default is once per day [`daily`] but you may also select once per hour [`hourly`].

Timezone NTP Server NTP Updates

If your node is unable to connect to the NTP server specified here then it will briefly search for another NTP service which might be defined on your local mesh network. The node hosting that service must enter its Advertised Service with “NTP” as part of the service name. The protocol should be set to “ntp://”, the hostname should point to the host providing the service, and the port should be set to “123”, the standard NTP port. For example, `ntp://ab7pa-box2.local.mesh:123` would identify the NTP server portion of the Advertised Service. See the **Configuration Deep Dive** section for additional information about Advertised Services.

5.3 Next Steps

After you finish configuring your node and click *Save Changes*, your node will immediately reboot using your new configuration. Your node will have an IP address in the 10.x.x.x range, so you should set your computer to use [DHCP](#) to obtain a new IP address from your node. As explained in the installation checklists, you may need to disconnect/reconnect or disable/enable your computer's Ethernet interface so that it begins using the new IP address. You can open a web browser and enter `http://localnode.local.mesh` or `http://<your-nodename>.local.mesh` to login to your node.

Link: [AREDN Webpage](#)

NODE STATUS DISPLAY

Once you have completed the initial setup on your AREDN® node, you can connect your computer to the LAN port on the PoE and navigate to the following URL: `http://localnode.local.mesh`. You will be redirected to the **Node Status** page as shown below.

AB7PA-A752

Location: 33.4455 -110.7788

Desktop node with LAN AP enabled

Help Refresh Mesh Status WiFi Scan Setup Select a theme ▾

mesh address: 10.92.237.46 / 8	signal noise SNR: no RF links Charts
mesh gateway: 172.31.35.122	
gateway node: KI7LXY-HAP	firmware version: 3.23.12.0
SSID: AREDN-10-v3	model: MikroTik RouterBOARD 952Ui-5ac2nD (hAP ac lite)
channel: 178	
channel width: 10 MHz	antenna: 1.5 dBi Omni
frequency range: 5885 - 5895 MHz	system time: Fri Dec 8 2023 08:34:45 MST
LAN address: 10.116.135.73 / 29	uptime: 0:16
LAN AP SSID: AB7PA-AREDN	
WAN address: 192.168.10.5 / 24	load average: 0.25, 0.31, 0.27
default gateway: 192.168.10.1	available space: <i>flash</i> = 9640 KB
	<i>memory</i> = 23712 KB
	host entries: 87 nodes / 207 total devices

Part of the AREDN™ Project. For more details please [see here](#)

Below the node name bar there are several controls.

Help

Opens a new window or tab to display the node help page.

Refresh

Updates the Node Status page with current data.

Mesh Status

Opens the **Mesh Status** page showing the neighbor nodes and remote nodes visible on the mesh network, as well as what services are being provided by those nodes.

WiFi Scan

Displays a list of other 802.11 signals within range of your node. The 802.11 signals may include Access Points, neighbor nodes, and other mesh networks (foreign ad-hoc networks). WiFi Scan is described in more detail below.

Setup

Navigates to the **Setup** pages for your node. You will need to supply a username and password to access those pages. The username is always `root`, while the password is the one you set during initial node setup. If the node has not yet been configured, the password is `hsmm`.

Select Theme

AREDN® firmware has several built-in display themes. The default `aredn` theme has a gray background with black and red text. The `black_on_white` theme is often chosen because it provides the best screen contrast on a computer exposed to direct sunlight. `red_on_black` is much better suited for nighttime use since it helps preserve night vision.

6.1 Node Settings Summary

The area under the control buttons shows both configuration and network status information. The left column contains the IP address and gateway details for the network interfaces on this node, as well as the SSID, channel, channel width, and frequency range if Mesh is enabled. If WAN Wifi Client is enabled it will also show the SSID and signal strength to the connected Access Point. If LAN AP is enabled then the LAN AP SSID will also be displayed.

The right column contains the Signal Strength readings and other attributes of your node. The **signal / noise / ratio** shows the strongest neighbor signal strength in dBm (decibels relative to one milliwatt) from all connected stations, and it is available only when the node is connected to a neighbor node via RF (Radio Frequency). Click these links for further information about [Signal to Noise Ratio](#) and values measured in [decibels](#). There are many factors that impact the network throughput you can expect to achieve, but as a general rule the higher the Signal-to-Noise ratio the better the throughput for your RF links.

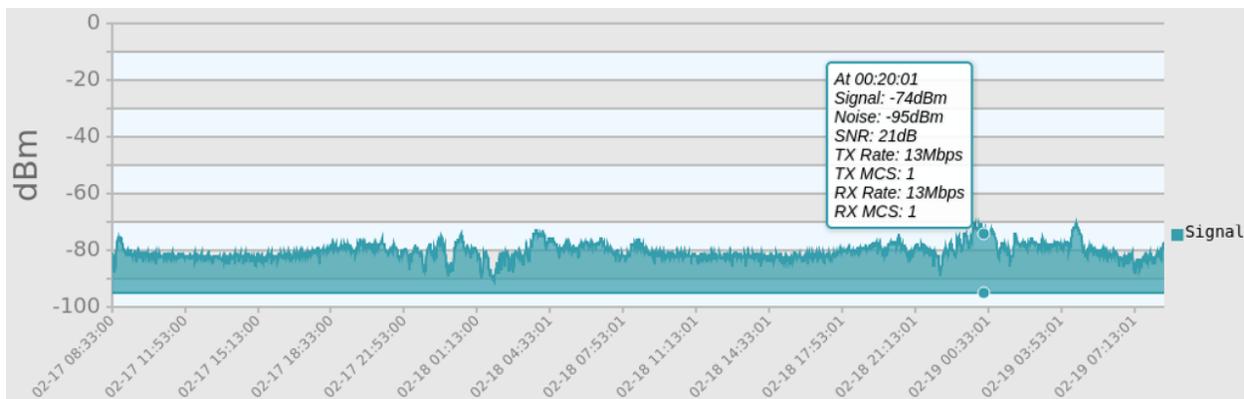
Below the Signal Strength readings are the node's **firmware version**, hardware **model**, and **antenna** info. The **system time** is displayed, as well as the **uptime**, which is the time since the last reboot. If an Internet connection or a local NTP (Network Time Protocol) server is available, your node's NTP client will sync its time with that time source.

The **load average** is the average number of processes that have been running on the node for the last 1, 5, and 15 minutes. **available space** tells you how much storage space is remaining on your

node. *flash* is the internal non-volatile storage where the operating system, configuration files, and software packages are kept. *memory* is the amount of RAM (Random Access Memory) available for running processes on the node. **host entries** shows the total number of devices seen on the network, and the total includes the AREDN® nodes as well as any other networked devices such as computers, VoIP phones, PBX devices, cameras, and other hosts.

6.2 Signal Charts

There is a **Charts** button next to the node’s **Signal Strength** display, and clicking this button takes you to **Signal Charts**. This page shows RF signal information in both a realtime and an archived view. The default view shows the average signal of all connected stations in realtime.



At the top of the charts display there are several control buttons.

Archive

This button shows the charts for any archived signal data on this node. Statistics are stored on the node in a circular buffer which holds about two days of data.

Realtime

This button shows the charts for current signal data as seen from this node.

Quit

This button exits the charts view and takes you back to the *Node Status* page.

Below these controls you can choose to view the signal strength statistics for individual nodes that are directly connected to your node. Choose the neighbor node from the **Selected Device** dropdown list. Changing the selected device will automatically reload the chart to show that node’s information.

Hovering over data points within a chart will show additional information for each data point, including Time, Signal, Noise, SNR (Signal to Noise Ratio), TX Rate, TX MCS (Modulation Coding Scheme), RX Rate, and RX MCS. If no traffic is being routed to the neighbor, the Rate and MCS values may be zero until data is available. An MCS value of zero may indicate non-802.11n encoding schemes (ie. 802.11a/b/g).

The small icon with three vertical dots in the upper right corner of the chart allows you to download a snapshot of the chart to a graphic file on your local computer (jpeg or png).

Data shown in the **Archive** charts is not stored in permanent memory on the node. The node will store approximately two days of archived data, and all data is cleared when a node is rebooted.

If you click and drag your mouse across a region of the chart, the display will zoom into that selected area. This allows you to view data points for a specific time range of your choice. While zoomed, two additional icons will appear in the upper right of the chart. The **Pan** icon allows you to scroll and pan the zoomed portion of the chart. The **Reset** icon returns the chart to its normal display mode.



On the left of the Realtime Graph there is an **SNR Sound** control. Clicking the *On* button will cause your computer to emit a tone that corresponds to the relative SNR level, with higher pitch tones indicating better SNR. This feature was added in order to provide an audio queue to operators in the process of aligning directional antennas. When your antenna reaches a position at which the highest pitch tone is heard you can lock it down without having to look at the signal graph display, knowing that you are receiving the best signal available. You can also adjust the tone pitch and volume with the sliders on the sound control.

6.3 WiFi Scan

WiFi Scan initiates a *passive* scan for wifi signals that are within range, but it only reports devices on the same channel width as your node. When installing a node at a new location it is best practice to scan on 5, 10, and 20 MHz channel widths to find all other 802.11 signals in range. This information will help you to pick a channel clear of interference. Several scans may be necessary to find all devices in range. When multiple ad-hoc networks are visible (using different SSIDs or channels), the ID of each 802.11 *network* is displayed but not the individual nodes.

A passive scan does not transmit probes, so there is no risk that unintended transmissions will interfere with radar stations on DFS channels. Automatic scan mode is available, but running a scan continuously is not recommended if the node is actively routing traffic. Even though the auto-scan is passive and only listens for other beacons across all channels, there is a risk of data loss on the assigned channel.

Attention: With some devices, a scan will momentarily disconnect the wifi from the mesh so the radio is available to perform the scan operation. It is recommended that you perform a scan when connected to the device in some other way.

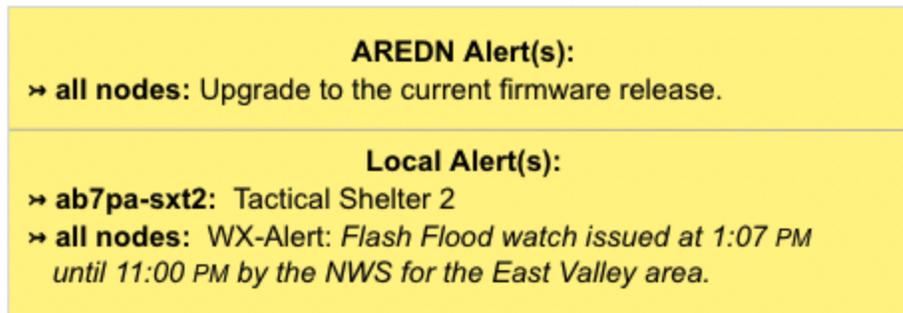
The scan results from your last scan are retained, along with the relative time since that scan was completed. If you only want to see the results from your last scan, you can go to the **Wifi Scan** page to view those results without having to initiate a fresh scan. Once a scan has finished, you can click the *Rescan* button to start a new scan. If you want your node to rescan continually you can click the *Auto* button. Click *Quit* to return to the **Node Status** display.

SNR	Signal	Chan	Enc	SSID	Hostname	MAC/BSSID	802.11 Mode
63	-49	178		AREDN-10-v3	AB7PA-SXT-1	DC:2C:6E:E2:D9:41	Connected Ad-Hoc Station
43	-69	178		AREDN-10-v3	AB7PA-Hub	AA:F6:26:65:9E:F3	My Ad-Hoc Network
43	-69	178		AREDN-10-v3	AB7PA-TEST2	2C:C8:1B:72:BC:62	Connected Ad-Hoc Station

Last scan: 14 seconds ago

6.4 AREDN® Alert Messages

AREDN® Alert Messages are displayed in a yellow banner on a node’s status page above the node name. There are three types of messages: broadcast messages intended for all nodes, group messages selected by labels in advanced settings and directed messages which are only retrieved by individual nodes. Individual nodes will attempt to pull the messages from the message repository URL once every 12 hours by default. Be aware that there is no guarantee of privacy for these messages, since anyone can view the message repository online. The AREDN® development team also has the ability to post messages which Internet-connected nodes will automatically retrieve.



The screenshot shows a yellow alert banner with two sections. The top section is titled "AREDN Alert(s):" and contains one item: "→ all nodes: Upgrade to the current firmware release." The bottom section is titled "Local Alert(s):" and contains two items: "→ ab7pa-sxt2: Tactical Shelter 2" and "→ all nodes: WX-Alert: Flash Flood watch issued at 1:07 PM until 11:00 PM by the NWS for the East Valley area."

Mesh nodes without Internet access also have the ability to display *Local Alerts*. The process for setting up a local message repository is described in the **Configuration Deep Dive** section, which allows node owners to decide whether to opt in to receive local messages on each of their nodes. If a node has Internet access as well as local messages, then both types of messages will be displayed in the AREDN® alerts banner as shown in the example above. There is also a web front-end application created by Gerard Hickey (WTOF) which runs directly on a node having adequate storage. You can find out more about this application in the **Applications and Services Guide** under the *Other Services* section.

[Link: AREDN Webpage](#)

MESH STATUS DISPLAY

The **Mesh Status** page lists mesh nodes and link quality information, along with any LAN hosts and advertised services available on the network. Below the node name bar there are several controls.

- The **Refresh** button refreshes the *Mesh Status* display with current information.
- The **Auto** button sets the display to automatically refresh the node information every 10 seconds. To end auto-refresh mode, click **Stop** or **Quit**. *Stop* returns to the static *Mesh Status* display. *Quit* takes you back to the *Node Status* display, but clicking *Mesh Status* again from there will return you to auto-refresh mode on the *Mesh Status* display.
- The **Cloud Mesh** button allows you to navigate to the *Mesh Status* display of the closest Supernode available to your device. Supernodes are a way to link multiple mesh island networks in a safe and efficient way. If your local node is part of a network with a Supernode, then you have the ability to view other nodes which are part of the Cloud Mesh network even if your local mesh is not otherwise linked to those networks. For further information see the *Supernode Architecture* description in the **Network Topologies** section of the **Network Design Guide**.
- The **Quit** button returns you to the *Node Status* display.
- The **Search** field allows you to filter the *Mesh Status* display by any keywords of your choice. The display will be limited to showing only nodes which match the keywords you enter. As you type each character from your keyboard into the search fields, the display will change to show only the entries that match your character or string. The filter is case insensitive, so it will find both upper and lower case entries for the characters you enter. If you press the **Refresh** button on the *Mesh Status* display, the search field will be cleared.

AB7PA-Hub mesh status

Location: 33.383 -111.505

Node Name	LAN Hostname							Service Name
AB7PA-Hub	ab7pa-voip2							Alert Manager Dial 10*231*105*114
Current Neighbors	LAN Hostname	LQ	NLQ	SNR	Quality	TxBps	Distance	Service Name
AB7PA-SXT-1 (rf,active)		100%	100%	53	100%	27.0		
KB7RJ-hAPac (tun,active)		100%	100%		100%		11 miles	
KI7LXY-HAP (tun,active)		100%	100%		100%		16 miles	
Previous Neighbors								
AB7PA-TEST2	8 minutes ago							
Remote Nodes	LAN Hostname							Service Name
KB7RJ-AR150 (tun*1)								
KB7RJ-B-O								KB7RJ: 10*195*90*131#
KB7RJ-NB5-S								
KB7RJ-NB5-W								
KB7RJ-PB2-SW								

7.1 Node Name

This shows your node as well as any connected LAN hosts and the advertised services available on your node and hosts. You can click any available web links to navigate to the services on your node or LAN hosts. This will be true for any available services in the *Current Neighbors* or *Remote Nodes* sections, too. Each node will be highlighted as you hover your cursor over it. This gives a visual indicator for any column entries that are part of the row over which you are hovering.

If you have any hosts for which you selected *Do Not Propagate* in the **DHCP Reservations List**, those hosts will be displayed in a light gray color only on your node's *LAN Hostname* column. If you created any **DNS Aliases** for your hosts, those aliases will be displayed in a light orange color only on your node's *LAN Hostname* column. All other hosts will be displayed in the default color for the theme that you are using.

7.2 Current Neighbors

This shows a list of *Neighbor Nodes* that are linked with your node. These nodes may be connected via radio, Device-to-Device link (dtd), a cross-link (xlink) or a tunnel (tun) over an Internet connection. The display also shows any LAN hosts on your current neighbors as well as any advertised services available on those nodes and hosts.

Link Quality Statistics

There are several link quality statistics displayed for each connected node.

- **LQ** or **Link Quality** is your node's view of the percent of **OLSR (Optimized Link State Routing protocol)** packets received from the neighbor node. These packets exchange mesh routing and advertised services information, and they include a sequence number that is used to identify missing packets. For example, if 7 of 10 packets sent by the neighbor were received, then the probability for a successful packet transmission from this neighbor is $7/10 = 0.7 = 70\%$. Be aware that the *Quality* metric is calculated differently, so there may not be a perfect alignment when comparing the two quality metrics.
- **NLQ** or **Neighbor Link Quality** is the neighbor node's view of the percent of **OLSR (Optimized Link State Routing protocol)** packets received from your node. This indicates the quality of the link from the neighbor's side.
- **SNR** or **Signal-to-Noise Ratio** is expressed in decibels (dB). It represents the level of signal which is detectable over the background noise floor, so a higher number is better. *SNR* is shown for both sides of any radio links (local SNR / remote SNR).
- **Quality** is the **Link Quality** calculated as the moving average of (total sent packets) divided by (total sent packets plus retransmissions), expressed as a percent. For example, if the node had to send every packet twice for it to be successfully received, the link quality would be 50%. An additional penalty is subtracted if the neighbor node is unpingable, which is explained in the *Advanced Configuration* section under "Ping Penalty". Be aware that the *LQ/NLQ* metrics are calculated differently, so there may not be a perfect alignment when comparing the two quality metrics.
- **TxBps** or **Transmit Megabits per Second** is an estimate of the data rate achieved across any radio (RF) link with a neighbor node. This column may show zero if the data being transmitted between these nodes is not sufficient for the metric to be calculated.
- **Distance** is the calculated distance between your node and each remote node. This calculation is based on the GPS coordinates (Lat/Lon) that were entered on each node. If no GPS coordinates were entered, then the distance cannot be calculated and that metric will not be considered in the LQM improvement process.
- **Service Name** is the column which displays any available services on the neighbor node or its LAN hosts. You can click on service links to navigate to the webpage for those services.

In addition to the neighbor node name, there are text abbreviations in parentheses that tell how the neighbor node is connected and the status of the link.

Link Type

- **rf**: indicates a radio link to this node.
- **dtd**: indicates a *Device to Device* connection (typically using an Ethernet cable) to this node.
- **wg**: indicates a Wireguard tunnel link over the Internet.
- **tun**: indicates a legacy Internet tunnel link.

- **xlink**: indicates a connection between the nodes that traverses cross-linked devices.

Link Status

- **wan**: the node has been configured as a *Mesh Gateway*. Typically this is a gateway to the Internet, but it may also be to another isolated network.
- **active**: LQM determined that the link is viable and is being used.
- **pending**: LQM is collecting data and evaluating the link.
- **idle**: LQM has determined that the link is usable and would be **active** but the node routing table does not yet have a route for sending traffic across the link.
- **blocked**: LQM determined that the link is unusable and has blocked it from use.
- **blocked - distance**: LQM determined that the remote node is either too close or too distant, based on the Min and Max Distance settings described in the *Advanced Configuration* section.
- **blocked - signal**: LQM determined that the SNR on the link is too low to reliably pass data, based on the Min SNR setting described in the *Advanced Configuration* section.
- **blocked - retries**: LQM determined that the retransmission rate is too high to reliably pass data.
- **blocked - latency**: LQM determined that the link latency is too high to reliably pass data.
- **blocked - dtd**: LQM blocks the RF interface on any nodes to which a DtD link also exists.
- **blocked - dup**: LQM blocks a link in cases when your node has an RF link to other nodes which themselves connect to each other via DtD. This can occur when there are multiple radios at a site using the same channel. The best remote node is chosen as the RF link for your node but the other possible RF connections are blocked as duplicates.
- **blocked - user**: LQM will block any node which you enter in the *User Blocked Nodes* field described in the *Advanced Configuration* section.

You can refresh the *Link Status* values by pressing the *Refresh* button or by selecting the *Auto* button to automatically refresh the display. Links whose quality has improved may be activated, while links whose quality has worsened may be blocked.

Previous Neighbors

If there were any Current Neighbors which disconnected within the last 24 hours they will be listed below any nodes that are currently connected. It shows the node name or IP address, as well as how long it has been since a node was actively connected to your node.

7.3 Remote Nodes

This section lists the other nodes on the network that are two or more hops away from your node. Advertised services on nodes and their LAN hosts are also listed. Remote Nodes are sorted by their ETX or *Expected Transmission* metric. ETX (Expected TX metric) is an estimate of the number of OLSR packets that must be sent in order to receive a round trip acknowledgement, and it is often referred to as *link cost*. When sending data the OLSR protocol selects the least cost route based on the lowest ETX in the direction of the final destination.

Link Status

- wan indicates the node has been configured as a *Mesh Gateway*. Typically this is a gateway to the Internet, but it may also be to another isolated network.
- (tun*?) indicates the node has tunnel links, with ? indicating the number of tunnels on that node.

[Link: AREDN Webpage](#)

CONFIGURATION DEEP DIVE

During your node's *Basic Setup* you used the configuration display by clicking the **Setup** button and typing your username and password. The configuration area has many additional features which will be described in more detail below. Clicking **Node Status** exits configuration mode without saving any changes, returning you to the *Node Status* display.



There are several control buttons below the configuration links section.

Help

Opens a new window or tab to display the node help page.

Save Changes

Click this button to save any configuration changes you have made. Saving changes will first do a basic validation of the new settings, saving them to flash memory if no errors are found. The new settings take effect in about 20 seconds and a reboot may or may not be required.

Reset Values

Click this button to reload the currently saved settings from flash memory, effectively undoing any changes that were made.

Default Values

Click this button to reset your node's basic settings to the default values. This action does not affect your existing node name.

Reboot

Click this button to force your node to reboot.

8.1 Basic Setup

You have already configured many of the basic settings, but there are several additional features that will be explained below.

Node Name	<input type="text" value="AD5BC-Node2"/>	Password	<input type="password"/>
Node Description (optional)	<input type="text"/>	Verify Password	<input type="password"/>

Mesh	LAN	WAN
Enable <input checked="" type="checkbox"/> Band <input type="text" value="5GHz"/> IP Address <input type="text" value="10.92.237.46"/> Netmask <input type="text" value="255.0.0.0"/> SSID <input type="text" value="AREDN"/> -10-v3 Channel <input type="text" value="36 (5180)"/> ? Channel Width <input type="text" value="10 MHz"/> <hr/> Power & Link Quality Tx Power <input type="text" value="22 dBm"/> ? Max Distance <input type="text" value="50.0"/> miles ? Min SNR <input type="text" value="15"/> dB Min Quality <input type="text" value="50"/> % <input type="button" value="Apply"/>	LAN Mode <input type="text" value="5 host Direct"/> ? IP Address 10.231.105.113 Netmask 255.255.255.248 DHCP Server <input checked="" type="checkbox"/> DHCP Start 114 DHCP End 118 <hr/> LAN Access Point Enable <input type="checkbox"/> ? AP band 2GHz SSID <input type="text" value="NoCall-AREDN"/> Channel <input type="text" value="1"/> Encryption <input type="text" value="WPA2 PSK"/> Password <input type="password"/>	Protocol <input type="text" value="DHCP"/> DNS 1 <input type="text" value="8.8.8.8"/> DNS 2 <input type="text" value="8.8.4.4"/> <hr/> WAN Wifi Client Enable <input type="checkbox"/> ? SSID <input type="text"/> Password <input type="password"/>

8.1.1 Mesh Column

Mesh is the node's *radio* interface. The AREDN® firmware has been designed to simplify the process of configuring networking interfaces. Network values are automatically calculated based on the unique MAC addresses of your node. You may need to change the *Channel* and possibly the *Channel Width* parameters to match those of your local AREDN® mesh, as explained previously in the **Basic Radio Setup** section. Normally you will not need to change the other network settings on this page, so keep these values unless you fully understand how the mesh works and why the defaults may not be suitable for your situation.

Channel Width Setting

Most AREDN® devices have a choice of using 20 MHz, 10 MHz, or 5 MHz channel widths. As a general rule, a larger channel width will allow more data to be transferred, but it may only do this over shorter distances. One suggestion is to start with the largest channel width that yields a *Signal to Noise Ratio* (SNR) of at least 15 dB.

Note: Some AREDN® devices will only support specific channel widths. If the choice of channel width is limited, the device will only show its supported widths in the *Channel Width* dropdown selector.

There may be several reasons why you might want to reduce the *Channel Width* setting:

- To achieve a better SNR on a marginal link.
- To extend the usable distance between neighbor nodes.
- To increase the number of available channels in a crowded RF coverage area.

Please review the **Network Design** section for more information about designing a network that meets the specific requirements of your applications and services.

Distance Setting

The screenshot shows a configuration panel titled "Power & Distance". It includes the following fields and controls:

- Tx Power:** A dropdown menu set to "23 dBm" with a help icon.
- Distance to FARTHEST Neighbor:** A text input field containing "0.00" with a unit selector set to "mi". Below it, another input field contains "0" with a unit selector set to "km".
- '0' is:** A text input field containing "0" with a unit selector set to "m".
- auto:** A radio button that is selected, with a horizontal slider below it.
- Apply:** A button at the bottom of the panel.

The *Distance* setting is only applicable to nodes that can communicate directly over RF. This setting adjusts the RF retry timer to define how long the transmitter will wait for an acknowledgement from a neighbor station. If the distance parameter is too short, the transmitter will send duplicate data packets before an acknowledgement has time to be received. If the distance parameter is too long, the transmitter will wait extra time before considering the data lost and retransmitting the packets.

Auto-Distance: A value of zero will cause the radio to automatically determine the RF retry timer by measuring the actual time it takes acknowledgement packets to be received. The timer is set using an Exponential Weighted Moving Average (EWMA). The auto-distance setting is best used on high quality, long distance point-to-point links between backbone or relay nodes. Fifty percent performance increases have been observed on those links compared to using a static distance setting.

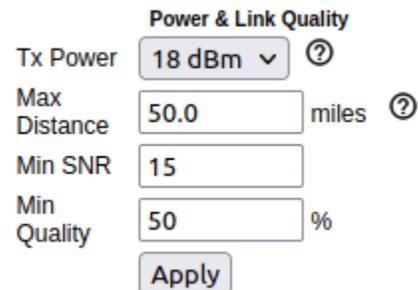
Since auto-distance causes the node to calculate the best value based on actual data flow, it will require both time and adequate data traffic to arrive at the optimal setting. The node may not be able to arrive at the optimal values if a link is not being used to send a significant

amount of data, because it starts at the max value and then drops down to the optimal setting. Over time the auto-distance setting should stabilize around the best value.

Attention: The auto-distance setting does **not** work well when nodes are in close proximity, when link quality is marginal, or when there are many nodes sharing the channel. In these cases the round-trip packet timing has a very wide range of values, so the timeout value becomes inflated and inconsistent. Static settings should be used in these situations.

A basic rule of thumb is when nodes are within five kilometers of each other you should test several *static* distance settings to see which one works best. The best way to test each distance setting is to use the **iperf3** package between endpoint nodes to measure the throughput of the RF channel under different distance settings. See *Test Network Links with iperf3* in the **How-To Section** for additional information.

Configuring LQM Settings



The screenshot shows a configuration panel titled "Power & Link Quality". It contains four input fields and an "Apply" button. The "Tx Power" field is a dropdown menu set to "18 dBm". The "Max Distance" field is a text input set to "50.0" with "miles" to its right. The "Min SNR" field is a text input set to "15". The "Min Quality" field is a text input set to "50" with "%" to its right. Each input field has a small question mark icon to its right. The "Apply" button is at the bottom of the panel.

When *Link Quality Manager* is enabled, the **Basic Setup** page will show a slightly different group of settings for *Power & Link Quality* under the **Mesh** column.

Max Distance

The maximum distance between nodes at which you can expect to achieve a usable radio link. The default value is 50 miles / 80 kilometers, but you can adjust this setting if your node is only able to maintain a usable radio link with closer nodes. Local conditions may dictate a shorter distance based, for example, on dense tree cover or other terrain features which impact line of sight communication. You can lower this value if you want to limit your node to linking only with nearby nodes.

Min SNR

The minimum Signal-to-Noise ratio that you require in order to reliably pass data between nodes. The default is 15 dB, but you can adjust this value if you require your node to continue passing data even on links that have reduced signal characteristics.

Min Quality

The minimum Link Quality required in order to reliably pass data between nodes. This is calculated as the moving average of total sent packets over total sent packets plus re-transmissions. For example, if the node had to send every packet twice for it to be successfully received, the link quality would be 50%. An additional penalty is subtracted

from Link Quality if the neighbor node is unpingable, and this is explained below under *Ping Penalty* in the *Advanced Configuration* section.

The **Power & Distance** settings can be adjusted and applied without saving changes or rebooting your node. However, they will return to their original values after a reboot unless you click *Save Changes*.

Enable/Disable Mesh

You can disable your node’s radio interface by deselecting the *Enable* checkbox, saving your changes, and rebooting the node. With the Mesh interface disabled the *Power & Distance* settings no longer apply and will be hidden. Since your node now has an unused RF interface, you will notice that a new section appears which allows you to use the node’s radio as an FCC Part 15 *LAN Access Point*. You can enable or disable the LAN AP using the *Enable* checkbox. See the details below for configuring the LAN Access Point.

Mesh		LAN	
Enable	<input type="checkbox"/>	LAN Mode	5 host Direct ?
IP Address	10.92.237.46	IP Address	10.231.105.113
Netmask	255.0.0.0	Netmask	255.255.255.248
		DHCP Server	<input checked="" type="checkbox"/>
		DHCP Start	114
		DHCP End	118
		LAN Access Point	
		Enable	<input type="checkbox"/> ?
		AP band	2GHz ?
		SSID	AD5BC-AREDN
		Channel	1 ?
		Encryption	WPA2 PSK ?
		Password	***** ?

8.1.2 LAN Column

The LAN column contains the settings for the Local Area Network hosted by the AREDN® node. There are several options under the *LAN Mode* dropdown.

The default mode is *5 Host Direct*. In this mode every host on the LAN has direct access to and from the mesh. This mode was created to reduce the amount of manual configuration needed to provide services to the mesh, since many services do not work well if they are hosted behind a NAT (Network Address Translation) router. With *Direct* mode the LAN shares the same address space as the mesh at large. Port forwarding is not needed because NAT is not used, and there is no firewall between the LAN and the mesh.

The mesh address space is automatically managed, so you cannot configure the LAN network settings in *Direct* mode. The only configurable option available in *Direct* mode is the size of the LAN subnet which can accommodate either 1, 5, 13, or 29 LAN hosts. A one host subnet can be used

for either a single server or a separate network router using its own NAT which is capable of more advanced routing functions than those available on a mesh node.

It is important not to use a subnet larger than is necessary because the chance of an IP address conflict on the mesh increases with the size of the subnet. The LAN subnet parameters are automatically calculated and depend on the IP address of the *Mesh* interface. If a conflict does occur it can be fixed by changing the *Mesh* IP address.

The other LAN Mode is NAT, and in this mode the LAN is isolated from the mesh. All outgoing traffic has its source address modified to be the *Mesh* IP address of the node. This is the same way that most home routers use an Internet connection, and all services provided by computers on the LAN can only be accessed through port forwarding rules. A single DMZ (DeMilitarized Zone) server can be used to accept all incoming traffic that is not already handled by other rules or by the node itself.

By default each node runs a DHCP (Dynamic Host Control Protocol) server for its LAN interface, which lets the node assign IP addresses automatically for devices connected to the node's local area network. The last octet of the start/end range for host IP addresses is shown in the LAN column. If you choose to disable the DHCP server, you must manually configure the host IP addresses to be within the LAN network range. There should be only one DHCP server for each IP address scope or range, so you may need to disable your node's DHCP server if there is already another device providing DHCP services on your node's local area network. Click this link for additional information on [Dynamic Host Control Protocol](#).

When you connect a device to your node's LAN, not only should it have an IP address in the LAN IP address range, but it is best practice for LAN devices to obtain their DNS Server information *automatically* from the node. Be aware that if a LAN device does not use the DNS Server entry provided by the node to which it is connected, then that device will be unable to resolve hostnames on the mesh network. Also, hard-coding a device's DNS Server entry with the mesh node's IP address could result in unexpected failures if that device is moved to another mesh node or network.

If you enabled the **LAN Access Point** feature mentioned previously, edit the access point's SSID, channel, encryption method, and password. Select an AP channel that is within the range supported by your WiFi client devices. Click *Save Changes* to write your information to the node's configuration, and a node reboot will also be required. Now wireless devices can connect to your node's LAN wirelessly, and their DHCP IP address will be assigned by the node's LAN DHCP server. If your node hardware has more than one unused radio, for example the *Mikrotik hAP ac* family with both 2.4 and 5.8 GHz radios in a single unit, the *LAN Access Point* section will always be visible whether or not your *Mesh* interface is enabled.

8.1.3 WAN Column

WAN

Protocol

DNS 1

DNS 2

WAN Wifi Client 

Enable 

SSID

Password 

The WAN (Wide Area Network) interface on your node is typically used to connect it to the Internet or to another external network. By default the WAN interface is set to obtain an IP address via DHCP from your upstream network. The DNS (Domain Name System) servers are set by default to use Google’s DNS services and should not be changed under normal circumstances. Google’s name resolution servers are configured properly to detect error conditions and report them correctly.

If you are not going to use the WAN interface on your node, you can select *disabled* from the *Protocol* dropdown list. If you will be using your node as a *Tunnel Server*, you should reserve an IP address on your router for the node’s WAN interface. This will be explained in the *Tunnel Server* section below. When a node has Internet access on its WAN interface, that access is available to the node itself and to any computers connected via the LAN port by default.

Note: The *Advanced WAN Access* settings have been moved to the **Advanced Configuration** display.

WAN WiFi Client

As mentioned above in the *Mesh* section, if your node has a radio on which you have *disabled* Mesh and you are not using it as a LAN AP, you can enable this available radio as a WAN interface by checking the **WAN WiFi Client** checkbox. Enter the SSID and authentication string for the wifi AP that you want to connect through for Internet access.

The mesh node uses “WPA2 PSK” encryption to connect to the wifi AP. The password length must be between zero and 64 characters. If the key length is 64, it is treated as hex encoded. If the length is 0, then no encryption will be used to connect to an open AP. A single quote character must not be used in the passphrase.

To the right of the *WAN WiFi Client* label is an icon with hover text indicating the status of the WAN WiFi connection.  indicates no wifi connection to the local access point.  indicates a wifi connection but no Internet connection.  indicates both a wifi connection

to the local access point and a connection to the Internet.

After you *Save Changes* and reboot, the node will have Internet access via wifi rather than requiring a cable plugged into the node's WAN port. In fact, enabling the *WAN Wifi Client* will disable VLAN1, so Internet access will no longer be possible through the physical WAN port. Also, on the *Node Status* display you will see the **WiFi WAN Address** label and IP address to indicate that your WAN connection is using the WAN WiFi Client.

8.1.4 Node VLANs

Many of the devices used as AREDN® nodes have only one Ethernet port, but more than one type of network traffic must share that single port. The AREDN® firmware implements VLANs (Virtual Local Area Network) in order to accomplish this. Different types of traffic are tagged to identify the network to which they belong.

VLAN 1

Packets received by the node that are tagged for VLAN 1 will be identified as WAN traffic from the Internet or another external network.

VLAN 2

Packets received by the node that are tagged for VLAN 2 will be identified as traffic from a DtD (Device to Device) node directly connected via Ethernet cable.

No VLAN tag

Packets received by the node that are untagged will be identified as LAN traffic from computers on the local area network.

It is important to understand AREDN® VLANs when configuring network smart switches for Internet access, tunneling, or DtD linking of nodes. There are some useful tutorials available on the AREDN® website for configuring VLAN-capable switches: [Video](#) or [Text+Images](#). Also, on the AREDN® GitHub site there is more information about node VLANs that have been preconfigured in the firmware images for specific types of radio hardware. For additional information visit this link: [Ethernet Port Usage](#)

8.2 Port Forwarding, DHCP, Services, and DNS Aliases

Click the **Port Forwarding, DHCP, and Services** link to navigate to these settings. This provides a way for you to configure LAN network address reservations and service advertisements on your node. The page works differently based on the LAN Mode (Direct or NAT) that you are using on your node.

8.2.1 Direct Mode Operation

DHCP Address Reservations					Advertised Services					
Hostname	IP Address	MAC Address	Do Not Propagate		Name	Link	URL			
ab7pa-srv	10.27.140.100	00:11:22:33:44:55	<input type="checkbox"/>	Del	MeshChat	<input checked="" type="checkbox"/>	http://ab7pa-srv	:	80 / meshchat	Del
ab7pa-voip	10.27.140.101	01:02:03:04:05:06	<input type="checkbox"/>	Del	MeshMail	<input checked="" type="checkbox"/>	http://ab7pa-mail	:	8080 /	Del
ab7pa-aux	10.27.140.98	07:08:09:10:11:12	<input checked="" type="checkbox"/>	Del	VoIP 10.27.140	<input type="checkbox"/>	://ab7pa-voip	:	/	Del
	- IP Address -		<input type="checkbox"/>	Add		<input type="checkbox"/>	://AB7PA-AR750	:	/	Add

Current DHCP Leases
there are no active leases

Port Forwarding					DNS Aliases	
Interface	Type	Outside Port	LAN IP	LAN Port	Alias Name	IP Address
WAN	TCP		- IP Address -		ab7pa-mail	ab7pa-srv
						- IP Address -

In Direct mode the LAN hosts are directly accessible from the mesh since no address translation or port forwarding is involved.

DHCP Address Reservations

If your node has its DHCP server enabled, it will automatically provide IP addresses to connected hosts. Look under the **Current DHCP Leases** heading to see the existing hosts and their assigned IP addresses.

Attention: The hostnames of every device connected to the mesh at large must be unique. It is best practice to prefix your Amateur Radio callsign to the hostname of each of your devices in order to have the best chance of it being unique on the mesh network.

Since DHCP leases are dynamic and can change over time, there may be a reason why a host's assigned IP address should be made permanent. This is especially useful if that host will provide an application, program, or service through your node to the mesh network at large. You can permanently reserve that host's DHCP address by clicking the *Add* button at the right of the row in the *Current DHCP Leases* list. You will see that host now appears in the list under the **DHCP Address Reservations** heading above the list of leases.

There may be some devices on which you are not able to set the hostname prefixed by your callsign. Once you add that device to your **DHCP Address Reservations**, however, click the *Hostname* box to edit the hostname what will be propagated across the mesh network by your node. You may also want to assign a specific IP Address to the device by selecting it from the drop-down list. If you have a device which needs to be reachable on its host node, but which should not be accessed across the mesh network, click the *Do Not Propagate* checkbox

to prevent OLSR from propagating that information to the mesh.

Once you have entered the values for your DHCP Reservation, click *Add* to add it to the list. You may also remove an existing reservation by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your changes to the node's configuration.

Advertised Services

Advertised Services include the applications, programs, or functions that are available to devices on the mesh network. The purpose of the network is to transport data for the services which are being used. Network services may include keyboard-to-keyboard chat or email programs, document sharing applications, Voice over IP phone or video conferencing services, streaming video from surveillance cameras, and a variety of other network-enabled features.

Services can run on the node itself or on any of its LAN-connected devices. Remember that AREDN® nodes have limited system resources with which to run services, so installing add-on services directly on the node should be avoided because the node could become unstable if sufficient resources are not available for normal operation, particularly on devices with only 32 MB of memory. It is best practice to run services on an external computer connected to the node's LAN network. In the example above you can see that an external host has been given a reserved DHCP address, and it is also running the *MeshChat* program as a service that is advertised on the network through this node. Use the following steps to create an Advertised Service.

Name

Enter a service name in the *Name* field.

Link

Check this box if you want your advertised service to display an active link in the web browser. This allows mesh users to navigate to your service by clicking the link in their web browser.

Protocol

Enter the protocol to use in the field between *Link* and *URL*. Common protocols include `http` for website services and `ftp` for file transfer services. Other services may use other protocols.

URL

From the dropdown list select the node or host on which this service is running. If you defined DNS Aliases as described below, you can also select a host alias from the dropdown list.

Port

Enter the network port on which the host is listening for service connections. There may be several applications provided through a single web server on a node or host using a single port, and in that case a valid application *Path* must be entered after the port number (as in the example above). In other cases the network port alone uniquely identifies the application or program that is listening for user connections to that service. You can find additional information at the following link: [Network Ports](#).

Once you have entered the values for your advertised service, click *Add* to add the service to the **Advertised Services** list. You may also remove an existing advertised service by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your changes to the node's configuration. A reboot is not required, and your new settings should take effect within thirty seconds.

Service Advertisement Process

OLSR (Optimized Link State Routing) propagates service entries to other nodes across the network. Once every hour your node will verify that its own service entries are valid. Your node will **not** propagate services across the network if it finds any of these conditions:

1. The host is not pingable across the network
2. There is no service listening on the specified port
3. The HTTP link does not return a *success* status code
4. The package for this service is not yet installed

The node's *Advertised Services* list will still show the defined service (with an alert icon and hover text marking it as non-advertised), but your node will not actually *advertise* that service to the network. If the service URL becomes reachable in the future or if the dependent package is later installed, then your node will resume advertising the service across the network.

Port Forwarding

In Direct mode you will only be allowed to select the WAN interface so Port Forwarding is only meaningful for WAN-connected nodes. Enter the Outside Port being passed to your node from its upstream gateway, select a LAN host to process the requests, and enter the LAN Port on that host which is listening for those requests. Finally, click *Add* to add the port forwarding rule. You may also remove an existing rule by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your port forwarding changes to the node's configuration. More information can be found at this link for [Port Forwarding](#).

DNS Aliases

DNS Aliases provide a way for you to create a hostname alias for a services computer. This can be useful if you want a computer or device on your node's LAN network to be identified by something other than its actual hostname. Your DNS Alias will be propagated across the network even if the actual hostname has *Do Not Propagate* checked in its DHCP Reservation, allowing you to hide the actual hostname while still advertising the alias on the mesh.

To create an alias, enter an **Alias Name**. The alias should be prefixed with your callsign in order to follow the naming convention used when defining any unique host on the network. Then use the dropdown selector to choose the name or *IP Address* of the existing host for which you are defining the alias. Once you have entered these values, click *Add* to add the alias to the list. You may also remove an existing alias by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your changes to the node's configuration.

Once an alias is defined, the **DNS Aliases** become available for creating *Advertised Services*. This feature can be used for virtual domain email servers, virtual machine identifiers, virtual web site URLs, and many other services.

8.2.2 Advanced DHCP Options

Tags for Advanced DHCP Options				Advanced DHCP Options				
Set a Tag Named	When Client's	Matches		For Tag	Always	Send DHCP Option	With Value	
polycom	Vendor Class	Polycom	Del	polycom	<input type="checkbox"/>	101 (tzdb-timezone)	America/Phoenix	Del
polycom	MAC Address	00:04:F2:*:*	Del	polycom	<input type="checkbox"/>	66 (tftp-server)	ftp://user:pw@pbxftp.local.m	Del
cisco	Vendor Class	Cisco	Del	[any]	<input checked="" type="checkbox"/>	42 (ntp-server)	10.22.33.4,10.55.66.7	Del
tagname	-Parameter-		Add	cisco	<input type="checkbox"/>	150 (tftp-server-address)	10.123.45.6	Del
				[any]	<input type="checkbox"/>	- DHCP Option Number -		Add

The **Advanced DHCP Options** section allows you to specify option codes and values which are sent to devices on your node's LAN network at boot time. This provides an easy way to configure network clients during their boot process. In addition to providing an IP address, the DHCP protocol is able to send a large number of options for device configuration. Any LAN client device joining the network can request specific DHCP options in addition to its IP address. These *Advanced DHCP Options* are especially helpful for configuring and provisioning VoIP phones on your node's LAN.

The [Internet Assigned Numbers Authority \(IANA\)](#) is the source for information about all DHCP options. Specific vendor equipment may or may not support all of the options, so you can verify which options are supported by referring to the manufacturer's documentation for your LAN device.

Tags for Advanced DHCP Options

The *Tags for Advanced DHCP Options* table allows the administrator to define DHCP tags that will be assigned to clients which are identified by specific values or properties such as Vendor Class or MAC address.

Advanced DHCP Options

The *Advanced DHCP Options* table allows the administrator to specify DHCP options that will be sent to any client, or only to clients matching a specific tag. Option numbers can be entered directly or chosen from a list of well-known options. Option values are manually entered in the "with Value" field on each row.

Field data validation is implemented for any input field with easily recognizable content such as host name, MAC address, port and option numbers. Placeholders are also supplied for input fields that might otherwise be difficult to describe (such as MAC addresses) using wildcard matching. Once the appropriate values are entered, click the *Add* button to include the settings which were defined. You may also delete a row by clicking the *Del* button for that row. After you have added, changed, or deleted your Advanced DHCP Options, click the *Save Changes* button at the top of the page.

8.2.3 NAT Mode Operation

Port Forwarding					Advertised Services		
Interface	Type	Outside Port	LAN IP	LAN Port	Name	Link	URL
WiFi	TCP	8100	ab7pa-t430	80	AREDN docs	<input checked="" type="checkbox"/>	http://AB7PA-A75:8100/aredndocs
WiFi	TCP		- IP Address -			<input type="checkbox"/>	://AB7PA-A75:

DMZ Server:

DHCP Address Reservations			DNS Aliases	
Hostname	IP Address	MAC Address	Alias Name	IP Address
ab7pa-t430	172.27.0.13	d8:eb:97:b6:d0:37	ab7pa-softphone	ab7pa-t430
	- IP Address -			- IP Address -

Current DHCP Leases

ab7pa-t430	172.27.0.13	d8:eb:97:b6:d0:37
------------	-------------	-------------------

If you are using NAT for your LAN mode, then hosts on the LAN are isolated from both the Wifi and WAN interfaces by a firewall. This makes them inaccessible from either of these interfaces unless Port Forwarding is configured. In this mode all outgoing LAN traffic has its source address modified to be the Mesh IP address of the node. This is the same way that most home routers use an ISP Internet connection.

Port Forwarding

Port forwarding rules can redirect inbound connections from the Wifi, WAN, or both interfaces and forward them to an IP address and port on the LAN. The destination port need not be the same unless you are forwarding a range of ports as explained below.

To create a port forwarding rule, select the network *Interface* on which the traffic will enter your node. Select the *Protocol Type* used by the incoming packets (TCP, UDP, or Both). Enter the *Outside Port* number that the external request is using to connect to your service. When your node receives traffic on the selected interface, protocol, and port then that request will be routed to the *LAN IP* address and *LAN Port* of the host which is listening for incoming requests for that service.

Once you have entered these values, click *Add* to add the rule to the **Port Forwarding** list. You may also remove an existing rule by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your port forwarding changes to the node's configuration.

Example:

On the LAN of a mesh node called `ad500-mobile` there is an IP camera with an IP address of `172.27.0.240` which is running its own web display. To make that camera available to everyone on the mesh, create a port forwarding rule on the Wifi interface whose *Outside Port* is any unused port on your node (for example `8100`) with an LAN

IP of 172.27.0.240 and LAN Port of 80. This takes all connections to port 8100 on ad500-mobile and redirects them to port 80 on 172.27.0.240. In a web browser on a remote computer connected to the mesh you could go to `http://ad500-mobile:8100` to view the IP camera.

If you want to forward a range of ports, the *Outside Port* field will accept a hyphen-separated range in the form “xxxx-xxxx”. When doing this, set the LAN Port to the low value of the port range.

If you want to forward every port that is not already in use to a single computer on the LAN, choose that host’s IP Address from the **DMZ Server** dropdown. There can be only one DMZ Server. Be aware that this bypasses the firewall in the node, so the DMZ server should run its own firewall to prevent unauthorized access.

Note that port forwarding to an FTP server, which uses both ports 20 and 21, can be done with a single rule using port 21 if the ftp client is capable of using passive ftp mode. Web browsers are able to do this and handle ftp downloads seamlessly.

Advertised Services

In NAT mode Advertised Services will not be accessible until at least one port forwarding rule or a DMZ server has been defined as described above. Advertised Services are entered as they are for Direct mode, except that the URL field is always that of your node which is handling network address translation. The port number should be the one used as the *Outside Port* in the forwarding rule through which the service will be accessed. In the last field you can enter an optional path if needed, such as the name of a specific folder on a web server or a directory on an ftp server.

Click *Add* to add the service to the **Advertised Services** list. You may also remove an existing service by clicking the *Del* button. Click the **Save Changes** button to write your changes to the node’s configuration.

DHCP Address Reservations

DHCP Address Reservations make a LAN device’s IP address permanent so it can be used consistently when defining Port Forwarding rules, and they are added the same way as in Direct mode. If a LAN device is currently connected and has been given an IP address by DHCP then it will appear under *Current DHCP Leases*. If you click the *Add* button next to the lease then it will be added to the DHCP Reservations list. You may also remove an existing reservation by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your changes to the node’s configuration. When using NAT mode the IP addresses of LAN devices are **never** propagated across the mesh, so the *Do Not Propagate* checkbox will not appear on this page.

DNS Aliases

DNS Aliases work differently in NAT mode. Aliases **cannot** be propagated across the mesh, and they **cannot** be used when defining an *Advertised Service*. They can only be used as an alternate name for a device on the nodes’ LAN.

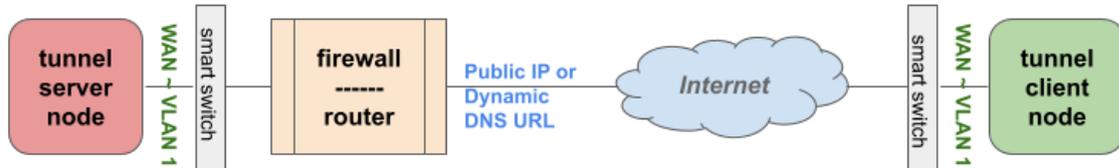
8.3 Tunnel Links

Tunnels are typically used as a means of connecting mesh islands if RF links cannot be established. Before using the AREDN® tunnel feature, be aware of how this type of connection could impact your local mesh network. If your node participates in a local mesh, then adding one or more tunnel connections on that node will cause the nodes and hosts on the far side of the tunnel(s) to appear on your local *Mesh Status* display. This adds complexity and makes everyone’s display a little more difficult to navigate. If you want to participate in remote mesh networks via tunnel, consider establishing those tunnels from one of your nodes that is *not* connected to your local mesh network. Also, remember that AREDN® is first and foremost an emergency communication resource, so it’s possible that Internet-dependent links and the assets they provide will not be available during a disaster.

8.3.1 Internet Connectivity Requirements

In order to run your node as either a *Tunnel Server* or *Tunnel Client*, you will need to configure Internet access. The following diagram shows an example of tunnel services between two nodes using the Legacy Tunneling Protocol described below.

AREDN® Tunnel Service Configuration



Node Settings:

- Static WAN IP address on the firewall/router network
- Create tunnel server credentials for each client needing a connection

Firewall Settings:

- Permit incoming traffic on port **5525** (*AREDN® default tunnel service port*)
- Any incoming requests for port **5525** should be forwarded to the static IP address of the tunnel server node on its port **5525**

Node Settings:

- No special settings required on firewall or router
- Enter the tunnel client credentials provided by the tunnel server owner

If you are using *Mikrotik hAP ac* family devices or *GL.iNET* devices then these nodes have built-in switches with the appropriate VLANs preconfigured in the AREDN® firmware. If you are using any other type of node, then you will need to configure a separate VLAN-capable switch. Set your VLAN-capable network switch to appropriately tag traffic from the Internet with “VLAN 1” before

sending it to your node. This allows your node to properly identify the traffic as coming from the Internet connection on its WAN interface. See the equipment manual for your smart switch to determine how to configure these settings.

8.3.2 Tunnel Server

Click the **Tunnel Server** link to navigate to these settings. This section provides a way for you to configure node-to-node connections across the Internet. The heading area displays information for both types of tunneling protocols. The legacy tunneling service provides an *unencrypted* connection between the linked nodes, while the Wireguard tunneling service provides an *encrypted* connection over the Internet. Tunnel network address ranges are calculated automatically and it is not necessary to change these settings unless there is a specific reason why the defaults will not work for your situation. The *Tunnel Server DNS Name* is the public IP Address or the *Dynamic DNS* name by which Internet-connected nodes can reach your network.

Tunnel Server Network: 172.31.163.252
 Wireguard Server Network: 172.31.164.252
 Tunnel Server DNS Name: 64.88.37.22

Allow the following clients to connect to this server:

Enabled?	Client	Pwd	Net	Active Action
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	172.31.163.252	<input type="button" value="Add"/>
Contact Info/Comment (Optional): <input type="text"/>				

Allow the following clients to connect to this Wireguard server:

Enabled?	Client	Key	Client	Active Action
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	172.31.164.252:5525	<input type="button" value="Add"/>
Contact Info/Comment (Optional): <input type="text"/>				

Legacy Tunneling Protocol

The top section is for entering tunnel clients for the AREDN® legacy tunneling protocol which uses TCP and is unencrypted. In the *Client* field enter the exact node name of the client node that will be allowed to connect to your tunnel server. Do not include the “local.mesh” suffix. In the *Client Password* field enter a password that the client node will use to connect to your node over the tunnel. Use only uppercase and lowercase characters and numbers in your password. You may also enter other optional information in the *Contact Info/Comment* field. To allow the client to connect to your tunnel server, select the *Enabled* checkbox.

Once these settings are correct, click *Add* to add the new client to the list of authorized tunnel clients. On the right of each entry there is an envelope icon which will automatically open

your computer's email program and copy the client settings into a new email which allows you to quickly and easily send credentials to the owners of the client nodes.

In order for your Internet-connected router/firewall to have a consistent way to forward traffic to your node, it is best practice to set a static IP address on your tunnel server node's WAN interface or to reserve its DHCP IP address in your router.

On your Internet-connected router/firewall set the firewall rules to permit TCP traffic from the Internet on port 5525. Then configure a port forwarding rule to send any traffic from the Internet on port 5525 to the IP address of your node's WAN interface.

Wireguard Tunneling Protocol

The bottom section of the *Tunnel Server* page is for entering tunnel clients that will use the Wireguard tunneling protocol which uses UDP and is encrypted over the Internet. In the *Client* field enter the exact node name of the client node that will be allowed to connect to your tunnel server. Do not include the "local.mesh" suffix. You may also enter other optional information in the *Contact Info/Comment* field. To allow the client to connect to your tunnel server, select the *Enabled* checkbox.

Once these settings are correct, click *Add* to add the new client to the list of authorized tunnel clients. The entry for the *Key* field will be auto-generated when the *Add* button is pressed. You will also see the port which was assigned to the entry in the *Client* field at the end of the IP address. On the right of each entry there is an envelope icon which will automatically open your computer's email program and copy the client settings into a new email which allows you to quickly and easily send credentials to the owners of the client nodes.

Note: If you change the *Client Name* on one of your existing Wireguard clients, the existing security key will be automatically retired and a new key will be generated. This may occur if the client node owner has changed its name, or if the Tunnel Server administrator needs to reuse/repurpose an existing line on the *Tunnel Server* display.

In order for your Internet-connected router/firewall to have a consistent way to forward traffic to your node, it is best practice to set a static IP address on your tunnel server node's WAN interface or to reserve its DHCP IP address in your router.

On your Internet-connected router/firewall set the firewall rules to permit UDP traffic from the Internet on an appropriate range of ports. The starting port should be 5525, which will provide for one wireguard tunnel connection. If you want to allow up to 10 wireguard tunnel links (for example), you would permit UDP traffic on the range of ports between 5525-5534. Then configure a port forwarding rule to send any traffic from the Internet on your range of ports to the IP address of your node's WAN interface.

Supernode Tunneling

Supernode tunneling uses the Wireguard tunneling protocol, but the port range begins with port 6526. On your Internet-connected router/firewall set the firewall rules to permit UDP traffic from the Internet on an appropriate range of ports. The starting port should be 6526,

which will provide for one supernode tunnel connection. If you want to allow up to 10 supernode tunnel links (for example), then you would permit UDP traffic on the range of ports between 6526–6535. Then configure a port forwarding rule to send any traffic from the Internet on your range of ports to the IP address of your node’s WAN interface.

Once the client information has been entered, click the **Save Changes** button. When a tunnel connection becomes active, the cloud icon at the right of each row will change to indicate that the tunnel is active. Depending on the timing of the webpage refresh, you may need to press the **Refresh** button to see the active icon.

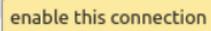
8.3.3 Tunnel Client

Click the **Tunnel Client** link to navigate to these settings. In this section you can configure your node to connect over the Internet to another node running as a *Tunnel Server*. You should already have your VLAN-capable network switch configured as explained in the *Internet Connectivity Requirements* section above, if it is needed.

Contact the amateur operator who controls the tunnel server and request client credentials by providing your specific node name. The tunnel server administrator will provide you with the public IP or DDNS (Dynamic Domain Name Service) name for the tunnel server, the password/key you are to use, and the network IP address for your client node. Enter these values into the appropriate fields on your node and click *Add* to create a client entry in the list.

Connect this node to the following servers:

Enabled?	Server	Pwd	Network	Active	Action
<input type="checkbox"/>	ab7pa.dynamicDNS.com	mySecretPassword	172.31.67.89		Del
Contact Info/Comment (Optional):					



If your tunnel server administrator used the envelope icon to create an email to send you the credentials, you can simply highlight/select the credentials from the email, copy the selection, and then paste that selection into any of the blank fields for a new Tunnel Client row. Your node will correctly populate each of the separate fields with the credentials you were sent.

To allow your client to connect to the tunnel server, select the *Enabled* checkbox and click the **Save Changes** button. When a tunnel connection becomes active, the cloud icon at the right of each row will change to indicate that the tunnel is active. Depending on the timing of the webpage refresh, you may need to press the **Refresh** button to see the active icon.

8.4 Administration

Click the **Administration** link to navigate to these settings. There are four sections that provide ways for you to manage the firmware, packages, security keys, and support data on your node.

Firmware Update

There are currently three ways to update the firmware on your node. No matter which method you choose, you can retain your existing configuration settings by selecting the *Keep Settings* checkbox.

- 1) **Upload Firmware:** If you have a new firmware image that you have already downloaded to your computer from the AREDN® website, click the *Browse* button and select the firmware file from the location on your computer where you saved it. Click *Upload* and the file will be uploaded and installed on the node.
- 2) **Download Firmware:** If your node has Internet access you can use the *Download Firmware* option. Click *Refresh* to update the list of available images. The source URLs that are queried are those listed on the *Advanced Configuration* page of your node. Select the image to download, click *Download*, and wait for the firmware to download and be installed.
- 3) **Load Local Firmware:** If you need to upgrade the firmware on a node which has a marginal connection to the network, the standard web/http method may not reliably transfer the image to the node. In this situation you may want to use an independent means of uploading the firmware to the node before beginning the upgrade process. Choose an upload method such as *scp* (secure copy) with a long connection timeout, which may allow the file transfer to continue the upload in the event of a network interruption. Transfer the new firmware file to your node, place it in the `/tmp/web` folder, and name it `local_firmware.bin`. Refresh your node's *Administration* page and once the page detects the `/tmp/web/local_firmware.bin` file, then the *Apply Local Firmware* button will become active. Press this button to begin the update process using the firmware you previously uploaded.

Package Management

Here you can install or remove software packages on the node. **Upload Package** allows you to install a package file by uploading it from your computer to your node. **Download Package**

allows Internet-connected nodes to retrieve a package from the AREDN® website. Clicking *Refresh* will update the list of packages available for download.

The **Remove Package** list shows all packages currently installed on the node. Selecting a package and clicking *Remove* will uninstall the package. You will only be able to remove packages that you have added. All installed packages are shown, but the pre-installed packages cannot be deleted since they are necessary for proper operation of the node.

Package Management

Upload Package	<input type="button" value="Choose File"/> No file chosen	<input type="button" value="Upload"/>
Download Package	<input type="button" value="- Select Package -"/> <input type="button" value="Refresh"/>	<input type="button" value="Download"/>
Remove Package	<input type="button" value="- Select Package -"/>	<input type="button" value="Remove"/>

As of NB 20230916, when you install extra packages, your node will remember them in its package store. When you next upgrade your node's firmware, the package store will be retained. After the firmware upgrade your node will wait for a few minutes and then automatically install the extra packages in its package store. If you *uploaded* the package to the node, then the package store keeps a copy of the package code itself. If you *downloaded* the package, then your node will attempt to redownload it. Also, if you later *remove* one of your extra packages, it will be automatically removed from the package store.

Authorized SSH Keys

Uploading ssh keys allows computers to connect to a node via ssh without having to know the password. The ssh keys are generated on your computer using built-in utilities or the PuTTY program's *Key Generator*. Once you have the key files on your computer, you can upload its *public* key to your AREDN® node. If you want to remove an installed key, select it and click the *Remove* button.

Authorized SSH Keys

Info: key file sanitized.

Upload Key	<input type="button" value="Choose File"/> No file chosen	<input type="button" value="Upload"/>
Remove Key	<input type="button" value="- Select Key -"/>	<input type="button" value="Remove"/>

Note: If you plan to use ssh keys you may want to review **Use PuTTYGen to Make**

SSH Keys in the **How-To Guide** section which describes this process in detail for users of Microsoft Windows computers.

Support Data

There may be times when you want to view more detailed information about the configuration and operation of your node, or even forward this information to the AREDN® team in order to get help with a problem. Click the *Download Support Data* button to save a compressed archive file to your local computer.

8.5 Advanced Network

If you have a supported multiport device (currently *Mikrotik ac2*, or *ac3* only), then you will see a menu option for **Advanced Network**. This provides a way for you to configure the ports on your multiport node. For more information on the AREDN® VLANs being used, refer to the *Node VLANs* description in the **Basic Setup** section above.

[Basic Setup](#)
[Port Forwarding, DHCP, and Services](#)
[Tunnel Server](#)
[Tunnel Client](#)
[Administration](#)
[Advanced Network](#)

Save Changes Default Values Reboot

Ports

	1	2	3	4	5
dtmlink vlan: 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
lan vlan: Untagged	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
wan vlan: <input type="text" value="Untagged"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Xlinks

VLAN	IP ADDRESS	PEER ADDRESS	WEIGHT	PORT	
20	172.16.1.1	172.16.1.2	1	3	<input type="button" value="+"/> <input type="button" value="-"/>

Ports

The **Ports** section shows the available ports across the top and the possible configurations along the left side. The default configuration is as follows:

- The first port is configured as a WAN port. The data entry field to the right of the *vlan* label can contain any valid vlan identifier if it is required, typically in the range between 1 and 4094. The default for these multiport devices is no vlan (untagged), so leave the default unless there is a specific reason why it is required in your situation.
- The middle ports are configured as LAN ports with no vlan (untagged).
- The last port is configured for DtD linking to another AREDN® node with vlan2 (tagged).

You should only have one box checked for each port. If you want to change a port's configuration, simply uncheck the existing box and check the box for the new setting on that port.

Xlinks

A cross-link allows your node to pass AREDN® traffic across non-AREDN® point-to-point RF links. To add a cross-link click the *plus* icon, enter an unused VLAN number for the link, the IP address of the near-side radio, the IP address of the far-side radio, a weighting factor, and the port to which the near-side radio is connected on your node. The *Weight* will be used by *OLSR* to determine the best route for AREDN® traffic. If you want to remove a cross-link, simply click the *minus* icon on the right side of the row to remove.

When you have finished making configuration changes to the ports and cross-links, click the *Save Changes* button. You will be notified if a reboot is required to activate your changes, and you can then click the *Reboot* button.

8.6 Advanced Configuration

The **Advanced Configuration** section allows you to change settings for various items that may be available on the type of hardware you are using. Not all hardware can support every value. These settings are best left as default unless you have a clear understanding of why you need to change the defaults for your node or network.

Above the settings table there are links that allow you to view the node help file, reboot the node, or reset the node to a firstboot or “NOCALL” configuration. You can edit or select a setting and then click the *Save Setting* button at the right side of the row to implement the change. You may also reset an item to default values by clicking the *Set to Default* button. For some settings you may need to reboot your node to apply the change, and in that case a message will be displayed notifying you that a reboot is required.

8.6.1 Link Quality Manager (LQM) Settings

Link Quality Settings		
<p>Enable Link Quality Management</p> <p>aredn.@lqm[0].enable</p>	<p>OFF <input checked="" type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>SNR Margin in dB above Min SNR a signal must reach to be re-activated</p> <p>aredn.@lqm[0].margin_snr</p>	<p>1</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Min Distance in meters beyond which a neighbor RF link is allowed</p> <p>aredn.@lqm[0].min_distance</p>	<p>0</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Default Distance in meters to use when actual distance cannot be calculated</p> <p>aredn.@lqm[0].auto_distance</p>	<p>0</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Quality Margin percentage increase before neighbor can be re-activated</p> <p>aredn.@lqm[0].margin_quality</p>	<p>1</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Ping Penalty quality percentage to add when neighbor cannot be pinged</p> <p>aredn.@lqm[0].ping_penalty</p>	<p>5</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>RTS Threshold in bytes before using RTS/CTS when hidden nodes are detected</p> <p>aredn.@lqm[0].rts_threshold</p>	<p>1</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Maximum packet size in bytes sent over WiFi (256 to 1500)</p> <p>aredn.@lqm[0].mtu</p>	<p>1500</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>User Blocked comma-separated list of blocked MACs</p> <p>aredn.@lqm[0].user_blocks</p>	<p></p>	<p>Save Setting</p> <p>Set to Default</p>
<p>User Allowed comma-separated list of always allowed MACs</p> <p>aredn.@lqm[0].user_allows</p>	<p></p>	<p>Save Setting</p> <p>Set to Default</p>

The basic LQM settings were described above under the **Mesh Column**, but additional LQM settings are also available here in the **Advanced Configuration** section.

Enable

Enable or disable the LQM feature in its entirety.

SNR Margin

The margin above the *Minimum SNR* that must be detected in order for a node to be returned to the active list based on signal level. The default value is 1 dB.

Minimum Distance

The minimum distance (in meters) that must exist between nodes in order for a link to be

considered for activation. The default value is 0 meters. This value can be increased if you do not want your node to pass traffic with nearby nodes, for example at a tower site with collocated backbone nodes designed to link only with other distant nodes.

Default Distance

The distance (in meters) to use when the actual distance between nodes cannot be calculated from their GPS coordinates. The default value is zero, which causes the node to treat nodes as being collocated.

Quality Margin

The margin above the *Minimum Quality* that must be detected in order for a node to be returned to the active list based on quality. The default value is 1 percent.

Ping Penalty

The Link Quality penalty that is imposed on calculations if a remote node does not respond to a ping request. The default value is 5 percent. This setting may be helpful for cases when a link would otherwise be marked *active* but the remote node is currently unreachable on the network.

RTS Threshold

The packet size in bytes triggering RTS/CTS when LQM detects hidden nodes. The default value is 1.

Maximum Packet Size

The maximum size of a packet which is sent over WiFi. The value is between 256 and 1500 with a default of 1500 bytes. Decreasing this value can improve link quality in some cases, especially in noisy environments with long distance connections.

User Blocked Nodes

A comma-separated list of MAC addresses which you desire to block from your neighbors list. This feature allows you to “blacklist” specific nodes. RF nodes are blocked by their Wifi MAC address, while DtD nodes are blocked by their LAN MAC address. MAC addresses are typically entered as uppercase characters with the hex pairs separated by colons.

User Allowed Nodes

A comma-separated list of MAC addresses which you always want to allow. This feature allows you to “whitelist” specific nodes. RF nodes are allowed by their Wifi MAC address, while DtD nodes are allowed by their LAN MAC address. MAC addresses are typically entered as uppercase characters with the hex pairs separated by colons.

8.6.2 WAN Settings

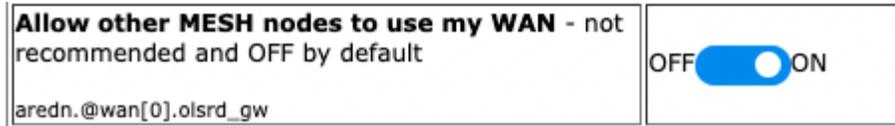
WAN Settings		
<p>Allow other MESH nodes to use my WAN - not recommended and OFF by default</p> <p>aredn.@wan[0].olsrd_gw</p>	<p>OFF <input type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Allow my LAN devices to access my WAN - ON by default</p> <p>aredn.@wan[0].lan_dhcp_route</p>	<p>OFF <input checked="" type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Provide default route to LAN devices even when WAN access is disabled</p> <p>aredn.@wan[0].lan_dhcp_defaultroute</p>	<p>OFF <input type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>WAN VLAN Number - must be an integer in the range [1,4094]</p> <p>aredn.wan.vlanid</p>	<p>1</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Enable web access to the node from the WAN interface</p> <p>aredn.@wan[0].web_access</p>	<p>OFF <input checked="" type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Enable SSH access to the node from the WAN interface</p> <p>aredn.@wan[0].ssh_access</p>	<p>OFF <input checked="" type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>
<p>Enable TELNET access to the node from the WAN interface</p> <p>aredn.@wan[0].telnet_access</p>	<p>OFF <input checked="" type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>

Several WAN access settings can be adjusted in this section. It is recommended that these settings be left at their default values, but specific use cases may require you to change them.

Allow MESH nodes to use my WAN

The default value is OFF and it is recommended that you use this default unless there is a special reason to enable it. Setting the value to ON will allow this node to route traffic from its Mesh interface to/from your WAN interface. Since the WAN interface typically provides a gateway to the Internet, it is not desirable to route Internet traffic over your Mesh interface. AREDN® is an FCC Part 97 amateur radio network, so be sure that any traffic which will be sent over the radio complies with FCC Part 97 rules. If you want local devices to have wireless Internet access, consider using an FCC Part 15 access point instead of your node’s WAN gateway.

In older firmware releases there was a checkbox on the *Basic Setup* display for this setting. In the past if you checked “Allow others to use my WAN” then here is what your slider would look like in the current firmware:

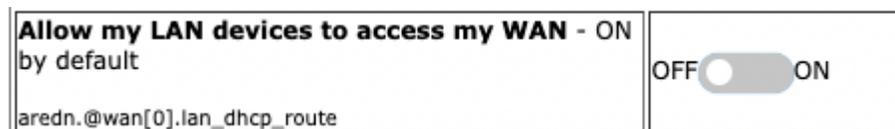


Remember that the default value is OFF and you should not turn it on unless you have a special use case.

Allow my LAN devices to access my WAN

The default value is ON which allows your LAN-connected devices to access your node's WAN network. Setting this value to OFF will prevent LAN devices from accessing the WAN, which means that your LAN hosts will not be able to reach the Internet even if your node has Internet access via its WAN. You may need to disable WAN access if your device needs to be connected to two networks at once, such as an Ethernet connection to your node as well as a WiFi connection to a local served agency network.

In older firmware releases there was a checkbox on the *Basic Setup* display for this setting. In the past if you checked "Prevent LAN devices from accessing the WAN" then here is what your slider would look like in the current firmware:



Remember that the default value is ON and you should not turn it off unless you have a special reason to do so.

Provide my LAN devices with a default route

Your node's DHCP server provides routes to LAN devices so they can access its available networks. A default route is required for WAN access, and that is provided automatically if "Allow my LAN devices to access my WAN" is ON as discussed above. However, some LAN devices (such as certain IP cameras) may not support DHCP option 121 and will require a default route in order to access the mesh network. Setting this value to ON will provide a default route to those devices. If a LAN device is connected to two networks at once, such as an Ethernet connection to your node as well as a WiFi connection to a local served agency network, care should be taken to understand how the device will deal with default routes to more than one network.

Remember that the default value is OFF and you should not turn it on unless you have a special reason to do so.

WAN VLAN Number

Important: This feature only applies to node hardware which requires a VLAN tag for the WAN interface. It will not appear on hardware where the Ethernet ports are on a switch chip, since changing the default VLAN number is not supported on those devices at the present time. It will appear as a blank field on devices that have a dedicated WAN port and therefore

do not need a VLAN tag for their WAN interface.

If you have node hardware that uses a VLAN tag for the WAN interface, then the default WAN VLAN identifier is 1. In some cases this default VLAN may be in use already or may be reserved by other equipment on your network. This field allows you to change the VLAN number being used on your node’s WAN interface.

Warning: If you plan to change this setting, do not use single digit identifiers or any number larger than can be supported by your network equipment. Different types of network equipment can support various numbers of VLANs, but the maximum number is limited by the [802.1Q standard](#) to no more than 4094.

Enable Web, SSH, or Telnet Access

HTTP, SSH, and Telnet access to your node is enabled by default on your node’s WAN interface. If you need to restrict this access to your node from the WAN, then you can turn it OFF here.

8.6.3 PoE and USB Power Passthrough

Power Options		
<p>PoE Passthrough specifies whether PoE power should be enabled (Not all devices have PoE passthrough ports)</p> <p>aredn.@poe[0].passthrough</p>	OFF <input type="checkbox"/> ON	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>USB Power Passthrough specifies whether USB power should be enabled (Not all devices have USB powered ports)</p> <p>aredn.@usb[0].passthrough</p>	OFF <input checked="" type="checkbox"/> ON	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>

These rows will only appear in the table if you have node hardware which supports PoE or USB power passthrough. One example is the *Mikrotik hAP ac lite* which provides one USB-A power jack, as well as PoE power passthrough on Ethernet port 5. You are allowed to enable or disable power passthrough on nodes with ports that support this feature. Move the slider to **ON** and click *Save Setting* to enable power passthrough.

8.6.4 Tunnel Options

Tunnel Options		
<p>Tunnel Weight specifies the cost of using a tunnel. The higher the number, the less likely a tunnel is used.</p> <p>aredn.@tunnel[0].weight</p>	<input type="text" value="1"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>WAN-Only Tunnel prevents tunnel traffic from being routed over the Mesh network itself</p> <p>aredn.@tunnel[0].wanonly</p>	OFF <input checked="" type="checkbox"/> ON	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>

Tunnel Weight

This specifies the OLSR route cost of using a tunnel, with the default value set to 1. The higher the route cost weight, the less likely a tunnel will be chosen for routing traffic.

Tunnel WAN Only Setting

This setting is enabled by default and it prevents tunnel traffic from being routed over the Mesh network. It limits tunnels to using the WAN interface, which is typically the intended route. If in your situation you need tunnel traffic to be routed over RF to a node with WAN access, then you can disable this setting to allow that traffic to pass.

8.6.5 Watchdog Settings

Watchdog		
<p>The Watchdog will reboot the node if it stops operating correctly</p> <p>aredn.@watchdog[0].enable</p>	OFF <input checked="" type="checkbox"/> ON	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>Watchdog IP addresses is a whitespace separated list of IP addresses, one of which should always be pingable</p> <p>aredn.@watchdog[0].ping_addresses</p>	<input type="text" value="8.8.8.8 10.200.31.5"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>Daily Watchdog hour is the hour every day (0-23) to automatically reboot the node</p> <p>aredn.@watchdog[0].daily</p>	<input type="text" value="2"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>

Watchdog

Watchdog is a background monitor that keeps track of core node processes. If any of the processes is having issues, *Watchdog* will reboot the node. This feature is OFF by default, but it can be enabled by moving the slide switch to the ON position and clicking the *Save Setting*

button. Currently the set of node processes that are monitored include olsrd, dnsmasq, telnetd, dropbear, uhttpd, and vtund (if tunneling is enabled).

Watchdog IP Addresses

You may also include one or more IP addresses, one of which should always be pingable. Your node will be rebooted if none of the IP addresses are reachable across the network. Enter IP addresses as a whitespace-delimited list.

Daily Watchdog Hour

Enter an integer between 0 - 23 which represents the hour of each day that you would like *Watchdog* to automatically reboot your node. The default is an empty field, in which case *Watchdog* will not auto-reboot your node.

8.6.6 Memory Settings

Memory Settings		
<p>Low Memory Threshold in KB when the Mesh Status page will be truncated</p> <p>aredn.@meshstatus[0].lowmem</p>	<input type="text" value="10000"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>Low Memory Max Routes is the maximum number of routes shown on the Mesh Status page when low memory is detected</p> <p>aredn.@meshstatus[0].lowroutes</p>	<input type="text" value="1000"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>

As the number of nodes increases in a mesh network, the processing requirements also increase for displaying all of the mesh routes on your node's *Mesh Status* display. For older nodes with limited memory resources, the mesh status display may become very sluggish on large mesh networks.

Recent firmware improvements have made the *Mesh Status* display much more responsive, but two **Advanced Configuration** values have been included for setting the *Low Memory Threshold* and maximum number of routes to be displayed.

Currently the default low memory threshold is 10,000 KB, which if reached will limit the *Mesh Status* display to the 1,000 closest routes. These values can be adjusted to lower values if your node has limited memory.

8.6.7 Supernode Settings

Supernodes are a way to link multiple mesh island networks in a safe and efficient way. If your local node is part of a network with a Supernode then you have the ability to view other nodes which are part of the Cloud Mesh network. This feature is ON by default and results in a new button being displayed on your *Mesh Status* page. The **Cloud Mesh** button will navigate to the *Mesh Status* display of the closest Supernode available to your device. For further information see the *Supernode Architecture* description in the **Network Topologies** section of the **Network Design Guide**.

Supernode Settings		
<p>Use any Supernodes found on the mesh</p> <p>aredn.@supernode[0].support</p>	<p>OFF <input checked="" type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>

Use any Supernodes

This switch enables or disables support for viewing remote networks connected through Supernodes. The default value is ON which means that your node will check for Supernodes and allow you to navigate to other networks via the **Cloud Mesh** button. Switching this value OFF will remove the **Cloud Mesh** button from your *Mesh Status* display.

8.6.8 Network Tools

Network Tools		
<p>OLSR Restart will restart OLSR when executed; wait up to 2 or 3 minutes to receive response</p> <p>aredn.olsr.restart</p>	<p>Click EXECUTE button to trigger this action</p>	<p>Execute</p>
<p>IPERF Enable allows the included iperf3 client/server</p> <p>aredn.@iperf[0].enable</p>	<p>OFF <input checked="" type="checkbox"/> ON</p>	<p>Save Setting</p> <p>Set to Default</p>

OLSR Restart

The **OLSR** (Optimized Link State Routing) process can be restarted when you want your node to rebuild its mesh routing table but you do not want to do a full reboot. Click the *Execute* button to restart OLSR.

There is a known intermittent issue that may occur when a node boots. If OLSR fails to propagate information or does not receive all the network hostnames, a one-time restart of OLSR should resolve the issue. OLSR should be restarted on your node if other nodes' *Mesh Status* display have your node's IP address rather than hostname or if "dtdlink" or "mid" is

shown in your node’s hostname on their *Mesh Status* display. If your node’s *Mesh Status* display shows the IP address rather than hostname for a remote node, then that remote node should restart OLSR.

iperf CGI Feature

The *iperf CGI* feature is described in the “Test Network Links with iperf3” section of the **How-To Guide**. It is enabled by default, but if you do not want your node to participate in any remote iperf tests then you can disable its ability to respond to those queries using this setting. Move the slider to OFF and click *Save Setting*.

8.6.9 Remote Logging URL

Remote Logging	
Remote logging URL for the remote syslog machine. Must be formatted as <i>protocol://ipaddress:port</i> aredn.@remotelog[0].url	<input type="text"/> <input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>

This field allows you to enter the URL for a remote syslog server. If this URL is provided, then your node will send log messages to the remote server using the specified IP address, port, and protocol.

8.6.10 Map Tile and Script Paths

Map Paths	
Map Tiles URL aredn.@map[0].maptiles	<input type="text" value="http://tile.openstreetmap.org/{z}/{x}/{y}.png"/> <input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Leaflet.css URL aredn.@map[0].leafletcss	<input type="text" value="http://unpkg.com/leaflet@0.7.7/dist/leaflet.css"/> <input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Leaflet.js URL aredn.@map[0].leafletjs	<input type="text" value="http://unpkg.com/leaflet@0.7.7/dist/leaflet.js"/> <input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>

These fields contain the external URLs for map tiles and *leafletjs css* and *javascript* files used for interactive maps.

8.6.11 Firmware and Package Download Paths

Firmware		
Firmware Download URL <small>aredn.@downloads[0].firmwarepath</small>	<input type="text" value="http://downloads.arednmesh.org/firmware"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Core Packages Download URL <small>aredn.@downloads[0].pkgs_core</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/targets/ath79/r"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Base Packages URL <small>aredn.@downloads[0].pkgs_base</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/packages/mips"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
AREDN Packages URL <small>aredn.@downloads[0].pkgs_arednpackages</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/packages/mips"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
LuCI Packages URL <small>aredn.@downloads[0].pkgs_luci</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/packages/mips"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Package Download URL for packages not included in the other sections <small>aredn.@downloads[0].pkgs_packages</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/packages/mips"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Routing Packages URL <small>aredn.@downloads[0].pkgs_routing</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/packages/mips"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Telephony Packages URL <small>aredn.@downloads[0].pkgs_telephony</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/packages/mips"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Freifunk Packages URL <small>aredn.@downloads[0].pkgs_freifunk</small>	<input type="text" value="http://downloads.arednmesh.org/snapshots/packages/mips"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
Dangerous Upgrade Disables all safety checks usually applied when upgrading firmware <small>aredn.firmware.dangerous_upgrade</small>	OFF <input type="checkbox"/> ON <input checked="" type="checkbox"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>

These fields contain the URLs used by the node for downloading firmware and package files during upgrades. By default they point to the AREDN® downloads server available across the Internet. You can change these paths to point to a local mesh package server in order to upgrade nodes that do not have Internet access. If you plan to create a local software repository for your mesh network, review **Creating a Local Package Server** in the **How-To Guide** section.

The **Dangerous Upgrade** setting allows you to disable the normal firmware compatibility safety checks that typically prevent you from loading the wrong firmware image on your node. The default setting is *OFF* which means that the safety checks remain enabled, and this setting should not be changed unless you have a specific reason to disable the firmware compatibility checks. One example for using this setting would be if you mistakenly installed an incorrect firmware image and

would like to correct that mistake by installing the correct firmware image (e.g., you installed the Mikrotik LHG version when you meant to install the LHG XL version).

8.6.12 AREDN® Alert Messages

AREDN Alert Settings		
<p>Alert Message Refresh - Execute to pull any AREDN Alert messages</p> <p>aredn.aam.refresh</p>	Click EXECUTE button to trigger this action	<input type="button" value="Execute"/>
<p>Alert Message Local URL - location from which local AREDN Alerts can be downloaded</p> <p>aredn.@alerts[0].localpath</p>	<input type="text" value="http://ab7pa-pi3.local.mesh/aam"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>Alert Message Groups - comma separated list of group names to check for alert messages</p> <p>aredn.@alerts[0].groups</p>	<input type="text" value="local-wx,fun-run"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>Alert Message Pollrate - how many hours to wait between polling for new AREDN Alerts</p> <p>aredn.@alerts[0].pollrate</p>	<input type="text" value="1"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
<p>Alert Message Purge - execute to immediately delete all alerts from this node</p> <p>aredn.aam.purge</p>	Click EXECUTE button to trigger this action	<input type="button" value="Execute"/>

Alert Message Refresh

The AREDN® development team may post messages which Internet-connected nodes can automatically download. You can execute the *aam.refresh* action if you want your node to retrieve any new messages without having to wait for the next auto-refresh window. Click the *Execute* button to trigger an immediate message retrieval. This will retrieve all alerts eligible for display on your node, whether they come from the AREDN® server over the Internet or from a local message source on your mesh network.

Alert Message Local URL

This field allows you to enter the URL for a local alert message repository. If you configure such a local repository then your nodes without Internet access can also receive alert messages pertinent to your local mesh. Enter the URL without a trailing backslash.

A local message repository can be configured on a mesh-connected web server which allows nodes to query the URL you entered. No Internet access is required for this feature to work. You can consult with your local server administrator in order to obtain the correct URL for the local message repository. You can find more information about AREDN® Alert Messages in the **Getting Started** guide under the *Node Status* section.

There is also a separate package called *AREDN Alert Message Manager* which allows the local message repository to be hosted on a node itself, rather than requiring a separate LAN-connected web server. You can find out more about this application by looking for *AREDN*

Alert Message Manager in the **Applications and Services Guide** under the *Other Services* section.

Use the following file naming convention on the web server:

- Create text files for individual node messages by using only lowercase characters with the exact node name, followed by the `.txt` extension as shown below. Whitespace characters are not allowed in node names.
- Create text files for group messages by using only lowercase characters with the group name, followed by the `.txt` extension. Whitespace characters are not allowed in group names.
- To create a broadcast message intended for all local nodes, enter your message text in a file named `all.txt` using only lowercase characters for the filename.

Index of /aam

Name	Last modified	Size	Description
 Parent Directory		-	
 ab7pa-sxt1.txt	2023-07-11 07:15	16	
 all.txt	2023-07-11 07:15	34	
 az-wx.txt	2023-07-11 07:10	109	
 fun-run.txt	2023-07-11 07:15	74	
 local-wx.txt	2023-07-11 07:14	70	

It is possible to include HTML tags in your message text, such as using the `
` tag to display subsequent text on the next line. However, it is best practice to keep alert messages short in order to minimize the height of the alert banner displayed on node webpages.

Alert Message Groups

In addition to local alert messages, it is possible to receive group alert messages. Group alert messages allow node operators to organize the mesh network into administrative/geographical domains or alert types using group labels. Multiple group names can be added to this field as a comma separated list.

Group alerts could be used by local operators to create a consistent alert structure. The following are some examples:

- Geographic regions (State, county, ARRL section, neighborhood)
- Connection types (backbone, leaf nodes, tunnels)
- Infrastructure *Change Management* notices
- Weather alerts
- Wildfire, flooding, tsunami or volcano alerts
- SKYWARN activations, DHS threat level

The group alert messages are retrieved from the web server specified in the local URL field. Alerts for a group are stored in a file named with the group name in all lowercase and a .txt extension as described above.

Alert Message Pollrate

This field allows you to set the polling rate or interval in hours at which the node will check for message updates. The default polling rate is once every 12 hours, but you can make this value smaller if you want your node to check for updates more frequently.

Alert Message Purge

Use this purge setting if you want to immediately remove the AREDN® Alert Message banner from your node. Click the *Execute* button to trigger an immediate message banner removal. This will remove all alert messages, whether they originated from the AREDN® server over the Internet or from a local message source on your mesh network.

8.7 Node Reset Button Actions

The reset button on an AREDN® node has two built-in functions based on the length of time the button is pressed.

With the node powered on and fully booted:

- **Hold for 5 seconds to reset the password and DHCP service**
- **Hold for 15 seconds to return the node to “just-flashed” condition**

On some equipment models it may be possible to accomplish these reset procedures by pressing the *Reset* button on the PoE unit.

[Link: AREDN Webpage](#)

REPORTING PROBLEMS OR ISSUES

If you experience issues with building or using AREDN® devices, there are several sources of help. There is an active user community that regularly contributes to the AREDN® [Forum](#), and you can post your experience there to receive help and feedback.

However, if you have issues that you think should be investigated by the AREDN® development team, you can follow the steps below for engaging with the software developers.

Download a Support Data File

Every node has a built-in tool that allows you to download a support data file containing information that is helpful for troubleshooting. To download a support data file from your node, navigate to the **Administration** page and scroll to the bottom. Click the *Download Support Data* link and your support file will be downloaded to your computer. If you are unable to navigate to the **Administration** page, you can simply enter this URL in your web browser to initiate the support data download:

`http://your-nodename-or-ip/cgi-bin/supporttool`



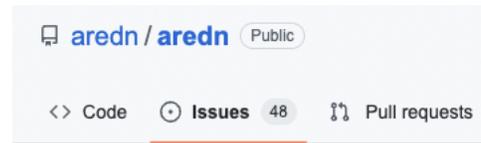
Create a GitHub account

To open an issue on GitHub you first must create your own GitHub account. This is free and easy to do by following these steps:

1. Open your web browser and navigate to the [GitHub URL](#).
2. Click the **Sign Up** button and enter the required information. We suggest using your callsign as the username.
3. On the GitHub website, click the **Sign In** button and authenticate to GitHub with the credentials you created.

4. Navigate on GitHub to the AREDN® code repository: <https://github.com/aredn/aredn>

Open a new issue on GitHub



There are several sections in the *aredn/aredn* code repository, and you can navigate to the issues area by clicking **Issues** in the top horizontal menu.

1. To open a new issue click the **New Issue** button on the upper right side.
2. Enter a meaningful title in the *Title* field.
3. Use the edit box to describe your issue fully. You should include the exact hardware model and firmware version on which you saw the issue.
4. You can attach screenshots or support data files by dragging and dropping them into the text window.
5. Click the **Submit New Issue** button to submit the issue for review.

Once the issue is submitted you can click the title in the issues list to see the details. You can enter additional information as a new comment on the existing issue. When any future comments or questions are posted to your issue you will receive notifications of those updates. If the issue has been resolved, you can then close your issue if you desire.

[Link: AREDN Webpage](#)

NETWORKING OVERVIEW

This **Network Design Guide** will discuss some of the useful principles for creating robust data networks as a service both to the amateur radio hobby and the community at large. An AREDN® network is able to serve as the transport mechanism for the applications people rely upon to communicate with each other in the normal course of their business and social interactions, including email, chat, phone service, document sharing, video conferencing, and many other useful programs. Depending on the characteristics of the implementation, this digital data network can operate at near-Internet speeds with many miles between network nodes.

There are a variety of ways to interconnect AREDN® nodes, but the most important question that should be answered is “*What is the purpose for this particular network?*” The specific requirements of your situation will drive the design of your data network. For example, consider the following issues.

Temporary or Permanent

Is your network being deployed as a short-term communication mechanism, possibly to meet the needs of a day-long event or a training exercise? If so, then several amateur radio operators with portable nodes can quickly establish an *ad hoc* mesh network with a specific set of services to meet the communication needs for that situation. Those nodes and computers can probably operate from portable batteries, without any external power dependencies for such a limited-time deployment.

Is your network intended as a long-term or permanent infrastructure to serve the on-going communication needs of a local region? If so, then a more sophisticated network topology must be designed and constructed to meet those long-term requirements. More robust or ruggedized radio equipment may be necessary, and more reliable AC power or off-grid renewable energy resources will be required to ensure consistent operations.

Geography and Terrain

Where is data communication needed? Are there specific locations where network nodes are required? What level of RF coverage will be needed in order to reach those locations? The places that the network must reach will determine the number and position of AREDN® nodes.

What are the geographical characteristics of the area across which your data network will operate? Different types of terrain may require specific types of network connections in order

to adequately cover the region over which data communications are needed. More demanding terrain may require a larger number of intermediate nodes or possibly larger higher-gain antenna systems and mounting structures.

Expansion and Growth

Will your network need to expand or adapt to changing conditions over time? Mesh networks are ideally suited for *ad hoc* growth and least cost routing based on the availability of nodes. As more devices are added to the network, however, a simple *ad hoc* mesh topology will not properly scale in size. It could result in increased latency on the network, with some network segments becoming almost unusable if application response time thresholds are exceeded. A growing network will probably require a different well-designed topology that routes data traffic efficiently in order to reach its intended destination.

Applications and Throughput

What network programs, applications, or services should be provided in order to fulfill the purpose for this network? Each application will generate a certain amount of data traffic, and some programs or services are more data-intensive than others. The network needs to be designed to adequately pass the traffic for the required applications.

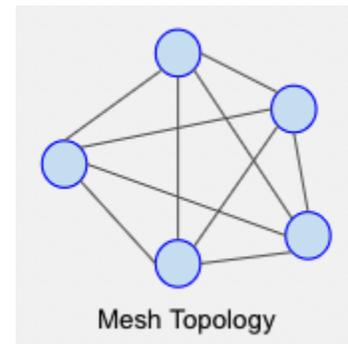
How many simultaneous users will be generating network traffic at different times? As the number of users increases, the amount of data traversing the network will also increase. In addition, with an increasing number of nodes on the network there will be a corresponding increase in the amount of [OLSR \(Optimized Link State Routing protocol\)](#) traffic that is necessary to maintain the network. An AREDN® network should be designed to handle the expected workload.

With these issues in mind, it is always best to keep your network as simple as possible and to include only those services which are required. Be sure to design your network so that it accomplishes its mission and suits its intended purpose.

[Link: AREDN Webpage](#)

NETWORK TOPOLOGIES

Every AREDN® node is capable of automatically joining an *ad hoc* mesh network which is operating with the same SSID, channel, and bandwidth. New nodes will each explore their surroundings by broadcasting their identity and listening for their neighbors' responses. Once nodes identify others within radio range, they share this information so that each node has a picture of the network topology. Periodic updates adjust the network routes based on changes in signal quality or loss of a link, allowing the network to adapt to changing conditions. Since there can be several possible routes between nodes, and since network disruptions typically effect only part of the network, a mesh topology can provide redundancy for network links.



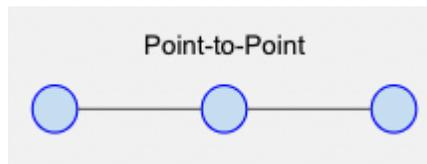
Every AREDN® node within radio range of other nodes will be able to participate in the network to extend its reach, provide route redundancy, or host services needed on the network at large. This simple mesh topology may serve its purpose perfectly for a short-term network deployed in support of a local event, or even for more permanent communication between nodes which are always within radio range. However, as mentioned in the previous chapter, the most important consideration for you network design is, “*What is the purpose for this particular network?*” The specific requirements of your mission should drive the design of your data network.

11.1 Types of Topologies

Although AREDN® nodes are capable of forming a simple mesh network, it is more common for operators to use different topologies in order to accomplish their data communication goals in growing networks. Typical network designs include Point-to-Point, Hub-and-Spoke, Tree or hybrid topologies.

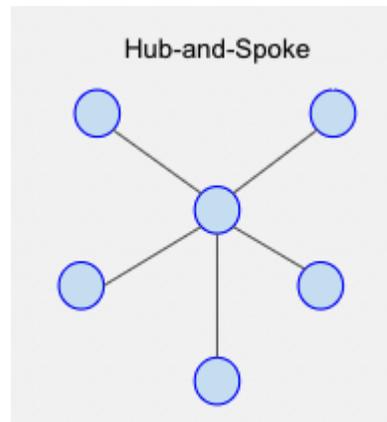
Point-to-Point Topology

Point-to-Point topologies are best suited for moving data between the far endpoints, potentially using one or more intermediate nodes in order to traverse different types of terrain or to overcome obstacles in the network path.



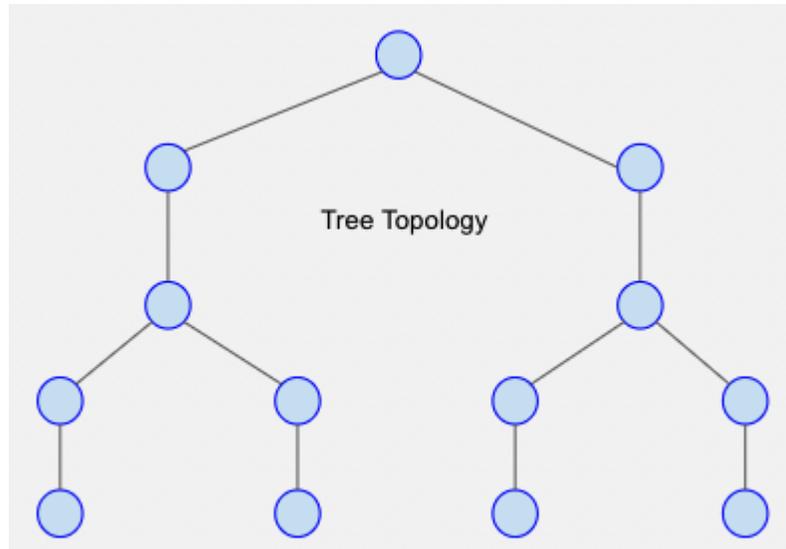
Hub-and-Spoke Topology

Hub-and-Spoke topologies work well in situations where the data communication to outlying nodes should be coordinated or funneled through a central location. Even if a remote node becomes unreachable, the rest of the network can continue to operate; but if the central node goes offline, the network will not function.



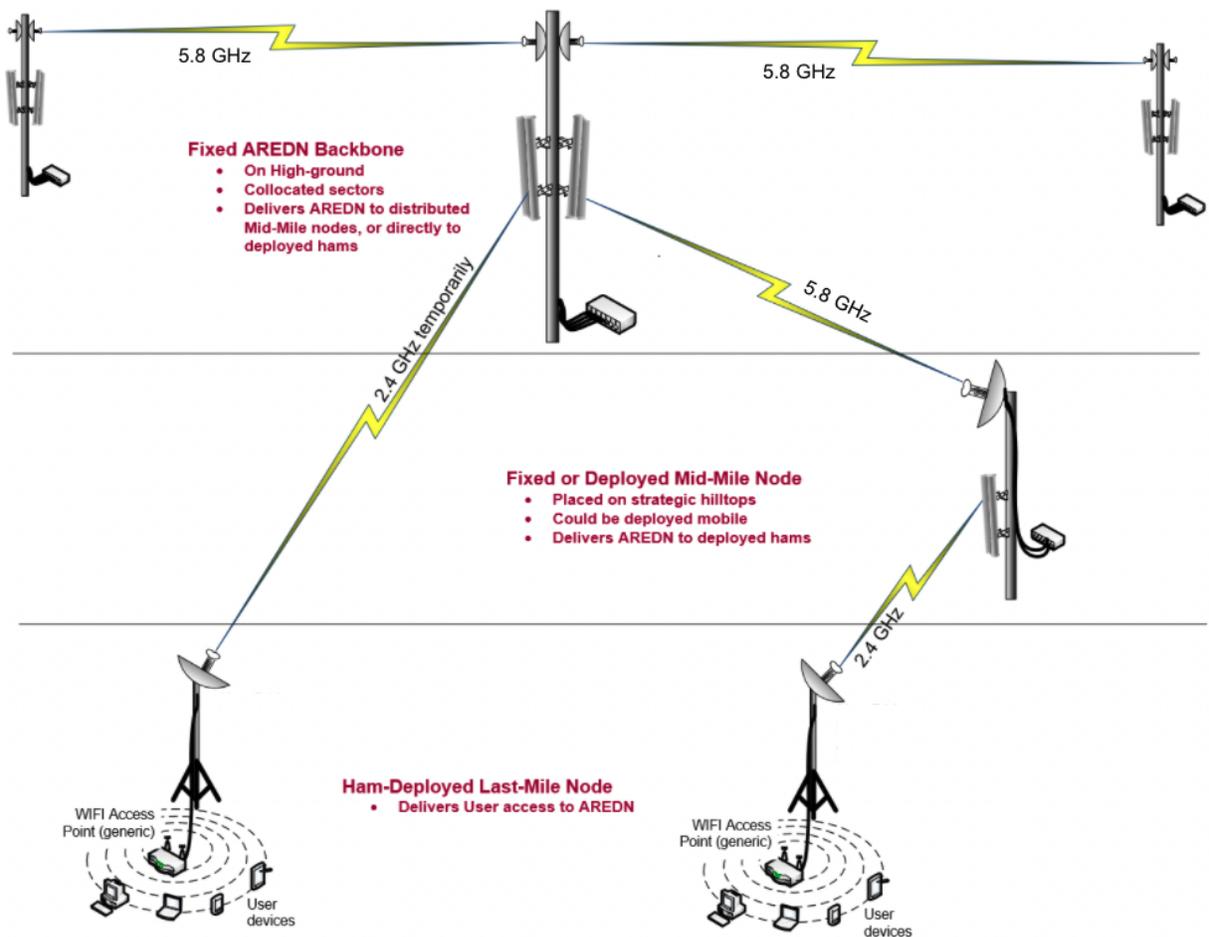
Tree Topology

A tree topology can be used to segment or partition network traffic, keeping specific data within a localized area while also allowing for links to remote parts of the network. The tree topology uses a parent-child hierarchy to structure the paths that data can take. This design can be easily scaled up or down to meet the specific requirements of the mission, but it does create “single points of failure”. If nodes go offline within the hierarchy then entire branches of the tree can become unreachable.



11.2 Types of Links

A *link* consists of both sides of a radio path, including the two devices that communicate back and forth across that path. Depending on the specific goals and the RF environment, there may be a need for special types of network links that connect the areas where data communication is required to fulfill your mission.



Backbone Links

As the name implies, these links form the backbone or superhighway along which large amounts of data can travel for long distances at relatively high speed. Typically backbone or “backhaul” links are permanent installations on mountain peaks, tall buildings, or high towers. They are usually point-to-point links with large high-gain antenna systems running on reliable power sources. In some cases these links are designed with redundant radios which help ensure path protection. Backbone links can operate over distances between 10 to 30+ miles.

Relay Links

Relay links bridge the gaps between endpoint nodes. Their primary purpose is to pass data efficiently, but there may be cases where they also serve as network access points for users. Sometimes these links are called “mid-mile”, “distribution”, or “intermediate” nodes. They are usually installed on medium-height towers or buildings in order to achieve high signal quality with good line of sight to other relay or backbone nodes. Depending on conditions, intermediate links may operate over distances between 3 to 10+ miles.

Endpoint Links

Endpoint links are used to connect destination nodes to the network. Sometimes these links

are called “last mile”, “tactical”, or “terminal” links. Usually the nodes at the far end will serve either as the originators or the final destinations for network traffic. Depending on local conditions, endpoint links typically operate over distances of 3 miles or less.

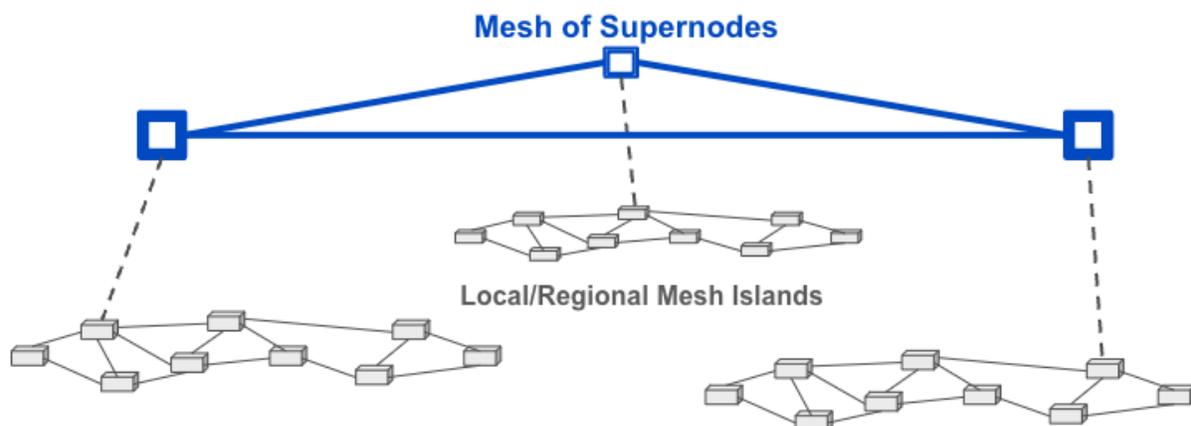
Different types of radio links may be needed to connect all of the nodes that are required in order to fulfill the purposes for your network. The ultimate goal of your network topology is to have a reliable data network that accomplishes its purpose for providing services to the intended destinations and users.

11.3 Supernode Architecture

Once several local or regional networks have been created, there may be a need for communication between these “mesh islands.” Often node owners have used direct Internet tunnel connections to accomplish this. However, this has the effect of merging the mesh islands into a single network with all of the routing traffic traversing all of the member networks. Many of the legacy nodes with older hardware/firmware are unable to handle the increased load.

A more efficient solution is to use a Supernode network to provide access across mesh islands, without sharing all of the local routing traffic across the linked networks. A Supernode is a specialized, dedicated node whose sole purpose is to link with other Supernodes and to shield each local network from the aggregate routing traffic. *Mikrotik hAP ac2* hardware is recommended for Supernodes, along with an Internet connection that provides robust bandwidth .

A Supernode network is a high-level mesh network — super meaning “*above or higher.*” The Supernode network sits above the isolated mesh networks and provides connectivity without increasing the routing load on the local networks.



A new solution for Supernode networks is currently being tested, and more information will be forthcoming in future documentation.

Link: [AREDN Webpage](#)

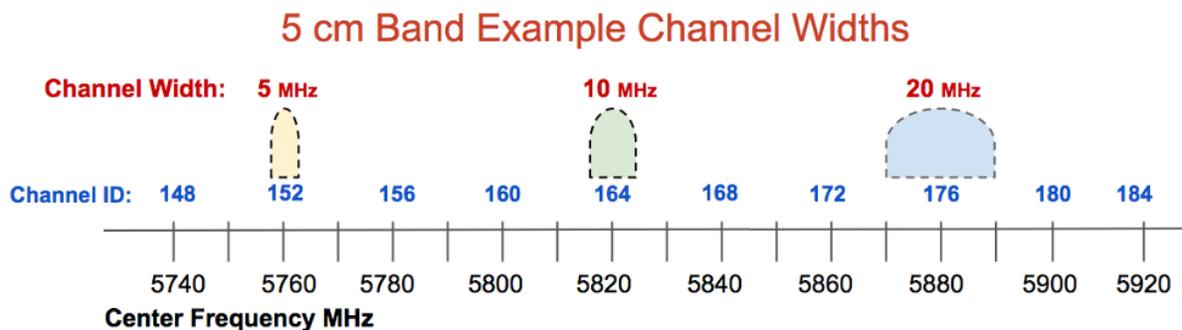
RADIO SPECTRUM CHARACTERISTICS

AREDN® networks operate in the microwave radio spectrum, and licensed Amateur radio operators have unique access to some of these frequencies. For bands in which Amateur operators share the spectrum, there is more chance for RF interference which may make some frequencies unusable for AREDN® data networking. For best results, select frequencies that are not being heavily used within the coverage area.

Attention: You are responsible for using frequencies, channels, bandwidths, and power levels that comply with your country’s Amateur radio license requirements.

Channel Information

Each band is divided into channels, each of which consists of a 5 MHz frequency offset identified by the center frequency of the channel and assigned a numerical label. In the example below you can see that a selected channel may use more or less of the frequency range based on the chosen channel width. The wider the channel, the more overlap there will be with adjacent channels. Wide channels have the effect of reducing the number of non-overlapping or non-interfering channels that will be available for use. When selecting channels and widths, be sure to use non-overlapping channels. Devices using channels or channel widths that overlap will interfere with one another and cannot communicate to coordinate the sharing of bandwidth.



Some or all of the bands shown below are shared with other authorized users. For example, all of the upper channels on the 13 cm band are shared with standard FCC Part 15 WiFi (IEEE 802.11x)

users in the US. The following table shows examples of the Amateur radio bands, frequency ranges, and number of channels that are available for AREDN® networking in the US.

Band	Frequency Range	Channels
5 cm	5650-5925 MHz	54
9 cm	3300-3445 MHz	14
13 cm	2390-2450 MHz	10
33 cm	902-928 MHz	4

The choice of a frequency band for AREDN® networking depends on several different factors, but you can “mix and match” bands in your network design as long as both sides of a radio link use the same band, channel, and channel width.

You have the option of selecting the channel width for each link. When using channels at the top or bottom of a band, be certain that your chosen width will not transmit outside of the FCC Part 97 allocation for that band. Different channel widths may yield better throughput than others. In some areas operators use different channels to isolate links, so they may need to use 10 MHz rather than 20 MHz channels in order to ensure they have enough available channels. Also, long distance links simply have better performance using 10 MHz vs. 20 MHz or 5 MHz channel widths. Test the performance of your links using various channel widths to ensure that they are optimized.

Power Limitations

The power limits that apply to AREDN® networks are the same as those that apply generally for Amateur radio operators in your country. As with any other operating mode, you should use the *minimum* power required to make radio links between nodes. In the United States, for example, this rule is specified in FCC part 97.313(a), and the maximum transmitter output power cannot exceed 1.5 kW PEP as specified by FCC part 97.313(b).

However there is one situation in the US where AREDN® devices are limited to 10W PEP. This special limitation applies to legacy devices that use 802.11b, which is a Spread Spectrum (SS) emission. FCC part 97.313(j) limits SS transmitter power to 10W PEP. All other AREDN® devices use 802.11n which transmits carrier waves with combinations of PSK and AM modulations. Refer to the 802.11n MCS rate tables for specific modulations that are used.

In actual practice, the output power of AREDN® devices will be limited by the hardware that is used. Even though in the US the FCC rules allow higher power, all of the modern commercial routers being used for AREDN® physically cannot transmit these high power levels. Therefore, the power limits allowed in the US by the FCC will never be reached unless you have an external Power Amplifier.

Some of the advantages and disadvantages of each frequency range are explained in the sections below which give examples of frequencies that are available to Amateur radio operators in the US.

12.1 5.8 GHz Characteristics

Advantages

One advantage for using the 5 cm band is that it contains 54 channels, and many of them may be under-utilized with less chance of interference. You can choose channel widths of 5, 10, or 20 MHz, with larger channel widths providing higher data rates. Remember that reducing the channel width may increase the SNR to improve signal quality if that is an issue for a marginal radio link.

The radio equipment and antenna systems for this band are readily available and are less expensive due to greater consumer demand. There is a wide variety of equipment from several manufacturers which supports the AREDN® firmware and operates across the 54 available channels. Radio and antenna systems for this band which are similar in size to those for other bands will often have higher gain. Devices in the 5 cm band are also well-suited for *Backbone Links* since there is little chance for RF interference from other radios sharing these frequencies at high profile sites. With clear line of sight and well-aligned antennas, 5.8 GHz signals can propagate across very long distances.

5.8 GHz	Channel	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148
	Ctr Freq	5.655	5.660	5.665	5.670	5.675	5.680	5.685	5.690	5.695	5.700	5.705	5.710	5.715	5.720	5.725	5.730	5.735	5.740
Status	Shared with US unlicensed indoor/outdoor DFS & Radar Avoidance															Shared with Unlicensed...			
	Channel	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166
	Ctr Freq	5.745	5.750	5.755	5.760	5.765	5.770	5.775	5.780	5.785	5.790	5.795	5.800	5.805	5.810	5.815	5.820	5.825	5.830
	Status	Shared with US unlicensed indoor/outdoor																	
	Channel	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184
	Ctr Freq	5.835	5.840	5.845	5.850	5.855	5.860	5.865	5.870	5.875	5.880	5.885	5.890	5.895	5.900	5.905	5.910	5.915	5.920
	Status	...Shared with Unlicensed		Shared with US unlicensed mainly indoor											Shared with Intelligent Transportation System				

Disadvantages

One concern with all of these frequency bands is that there must be clear line of sight between the nodes on each side of the link. This means that not only do the nodes need to have an unobstructed direct path, but the [Fresnel Zone](#) between the nodes must also be clear. The diameter of the Fresnel Zone depends on the frequency and the distance between nodes. If less than 20% of the Fresnel Zone is obstructed there is little signal loss, but any blockage beyond 40% will cause significant signal loss and could make the path unusable. For a link in the 5 cm band with 10 miles between nodes the first Fresnel Zone radius will be 46 feet, which is much less than the frequency bands discussed below. However, the 60% no blockage radius in the 5 cm band is still about 28 feet. Be sure to account for node AGL (height Above Ground Level) and terrain in order to achieve clear line of sight between nodes.

12.2 3.4 GHz Characteristics

Note: Late in 2020 the FCC ruled to sunset secondary Amateur allocations in the 9 cm (3.3-3.5 GHz) band. Although existing Amateur operations “*may continue while the Commission finalizes plans to reallocate spectrum,*” be aware that future FCC actions could remove Amateur operations altogether. Consider this before investing in or implementing new AREDN® devices in this band.

Advantages

Equipment in the 9 cm band is appropriate for *Backbone Links* since there is less potential for interference from other devices sharing these frequencies at tower sites. With clear line of sight and well-aligned antennas, 3.4 GHz signals can propagate across very long distances. You can select channel widths of 5, 10, or 20 MHz, with larger channel widths providing higher data rates. Remember that reducing the channel width may increase the SNR to improve signal quality if that is an issue for a marginal link.

3.4 GHz	Channel	76	77	78	79	80	81	82	83	84	85	86	87	88	89
	Ctr Freq	3.380	3.385	3.390	3.395	3.400	3.405	3.410	3.415	3.420	3.425	3.430	3.435	3.440	3.445
	Status	US Amateur operations remain on a secondary basis but are subject to removal at any time by FCC notice*													

* per FCC 20-138 IV-E-69

Disadvantages

Equipment for the 9 cm band is less readily available and is typically more expensive due to less consumer demand. Care must be taken when selecting radios so as not to confuse them with the more common WiMAX devices which are designed for the 3.65 GHz range and are not supported for use with AREDN® firmware. As mentioned previously, there must be clear line of sight and the Fresnel Zone between nodes also must be clear. For a link in the 9 cm band with 10 miles between nodes the first Fresnel Zone radius will be 62 feet, which is less than the 13 cm band discussed below. However, the 60% no blockage radius is still about 37 feet. Consider node AGL and terrain in order to minimize obstructions.

12.3 2.4 GHz Characteristics

Advantages

One advantage for the 13 cm band is that radio equipment and antenna systems are more readily available and less costly due to higher consumer demand. There is a wide variety of equipment from several manufacturers which supports the AREDN® firmware and operates in this band. With clear line of sight and well-aligned antennas, 2.4 GHz signals can propagate across very long distances.

Within the available frequency range you have the option of selecting channel widths of either 5, 10, or 20 MHz. A larger channel width will provide higher data rates. However, one effect of reducing the channel width is to increase the SNR to improve signal quality. For example, changing from a 20 MHz to a 10 MHz channel width will result in a 3 dB signal gain and could make the difference between a marginal link and a usable one. Just remember that when you cut the channel width in half you are also reducing your maximum throughput by half. Carefully test your links to ensure optimal performance.

2.4 GHz	Channel	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8*
	Ctr Freq	2.387	2.392	2.397	2.402	2.407	2.412	2.417	2.422	2.427	2.432	2.437	2.442	2.447
	Status	non-US only		Unshared		Cannot Use	Shared with US unlicensed							

* Only 5 MHz channel width is available on channel 8

Disadvantages

The upper channels of the 13 cm band are shared with several other FCC authorized services. Depending on local RF conditions it may not be possible to use these shared channels because of the high noise floor which reduces SNR and decreases signal quality. This leaves licensed Amateur operators only two unshared channels with a possible bandwidth of 5 or 10 MHz each.

As mentioned previously, there must be clear line of sight and the Fresnel Zone between nodes also must be clear. For example, on a link in the 13 cm band with 10 miles between nodes, the first Fresnel Zone radius will be 72 feet. In the 13 cm band the 60% no blockage radius is approximately 43 feet, which is often higher than most *Intermediate* or *Last Mile* nodes have been installed. Careful consideration must be given to node height and the terrain between nodes in order to minimize obstructions.

12.4 900 MHz Characteristics

Advantages

The advantage of this frequency band is that its longer wavelength makes it better suited for penetrating some types of foliage which would normally block signals at higher frequencies. Its NLOS (Non Line of Sight) propagation characteristics may be what is needed in order to establish an RF link between two difficult locations.

900 MHz	Channel	4	5	6	7
	Ctr Freq	907	912	917	922
	Status	Shared with US unlicensed			

Disadvantages

The entire 33 cm band is shared between several FCC authorized radio services. The disadvantage of using this band for AREDN® networking is that in all but the most remote areas the RF noise floor may be very high, which reduces the SNR and results in packet loss, retransmission delays, and lower usable link quality.

Another disadvantage is that the equipment can be more expensive than devices that operate in the 2.4 and 5.8 GHz bands. Also the entire band is quite narrow (25 MHz) which means that only one, two, or four radio channels can exist in this shared frequency range, depending on the channel width that is selected.

Different frequency ranges are available to connect the mesh nodes that are required in order to fulfill the purposes for your network. As you plan the frequencies to be deployed at specific locations, it may be helpful to use a *spectrum analyzer* for identifying ranges that are already in use. The ultimate goal is to have a reliable data network that accomplishes its purpose for providing services to the intended destinations and users.

[Link: AREDN Webpage](#)

CHANNEL PLANNING

The previous section identified the different channels in each frequency band which are available for AREDN® networking. Devices on each side of a radio link must use the same frequency band, channel, channel width, and SSID. Beyond that requirement, however, you have quite a bit of flexibility to select the radio channels that will ensure the highest signal quality and throughput for your network. In a basic AREDN® network with several nodes spread across a limited geographical area, all of the nodes may use the same band, channel, and channel width. This allows them to establish network routing to any of the sites as needed.

However, as more nodes join the network or when several nodes are COLLOCATED (same physical site) and share the same channel, it is possible for overall network performance to degrade. In order for an AREDN® network to scale up in size and complexity, frequency coordination and channel planning become increasingly important. To plan for future growth, local AREDN® groups may need to partition use different network topologies and to allocate different channels for specific geographic areas or types of links in order to ensure the network will be able to support the expected services.

13.1 Wireless Network Operation

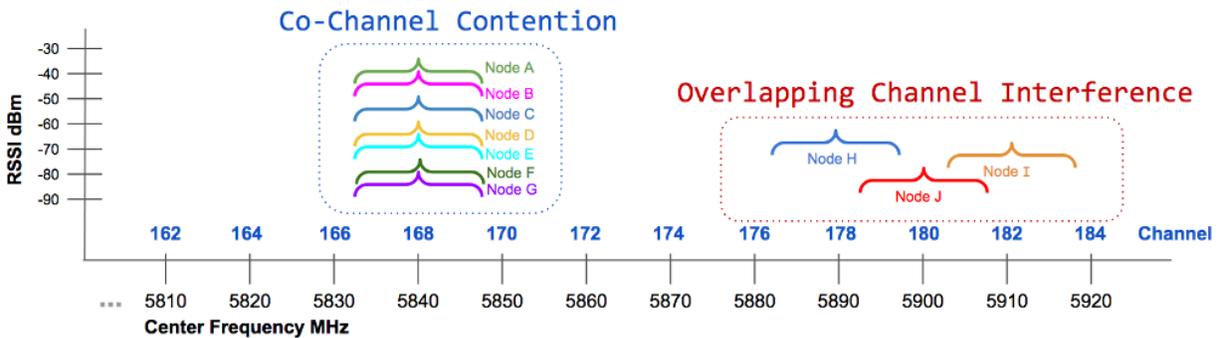
A wireless network is a shared half-duplex medium on which only one station at a time should transmit. In that sense wireless operations are analogous to other types of radio transmissions. If two stations key up their transmitters at the same time, they will interfere with each other to the extent that neither of them will receive the other's message. That is why net control procedures are implemented to ensure controlled access to a radio channel during emergency communication.

AREDN® firmware automatically mediates station access to the wireless medium by implementing IEEE 802.11a/b/g/n standards and Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA). This listen-before-talk technology helps nodes to determine whether a channel is busy. Each node performs a *Clear Channel Assessment (CCA)* as well as using *Request to Send / Clear to Send (RTS/CTS)* messages to negotiate access to a channel. A negligible amount of network traffic is also required for *OLSR (Optimized Link State Routing protocol)* to maintain routes for the network as a whole, but this OLSR traffic is a very small fraction of the total.

In a single-channel wireless network, any node that needs to transmit will automatically coordinate with the other nodes for a clear channel. This is by design, but as more devices try to gain access to the same channel there is an increased potential for each node to wait longer for its chance to transmit. This can result in increased latency and decreased network throughput as the number of network nodes increases.

13.1.1 Channel Contention

The concept of *Overlapping Channel Interference* is illustrated on the right side of the following channel scan diagram. *Overlapping Channel Interference* is very serious, but it can be eliminated by selecting non-overlapping channels for all of the devices accessing your network. A second issue related to how wireless networks operate is illustrated on the left side of the diagram. It is commonly called *Co-channel Interference* but is more accurately described as *Co-channel Contention* or *Co-channel Cooperation*.



In this example several nodes must share a single channel, so they all negotiate for the opportunity to transmit. Any node that needs to transmit will use listen-before-talk technology to determine whether the medium is busy. If the channel seems clear, the node will attempt to transmit data. If the channel is busy, the node will defer transmission until the channel seems clear. In a high-density network where a large number of nodes share a single channel, the normal negotiation processes may result in significant performance degradation. From an end-user perspective, an overloaded channel can make the network seem sluggish or even unusable.

This example is not meant to show that having only seven nodes will overload a channel. There is no established rule of thumb for channel sharing that specifies how many nodes are too many. The answer depends on the number of nodes, the bandwidth in use to support required services, the link signal qualities, and other network characteristics. Based on these parameters one shared channel may perform well with many dozens of nodes, while another network may see performance degradation with significantly fewer nodes sharing a channel. Many factors interact to influence network performance, but it will soon become obvious to users whether the network is behaving as expected.

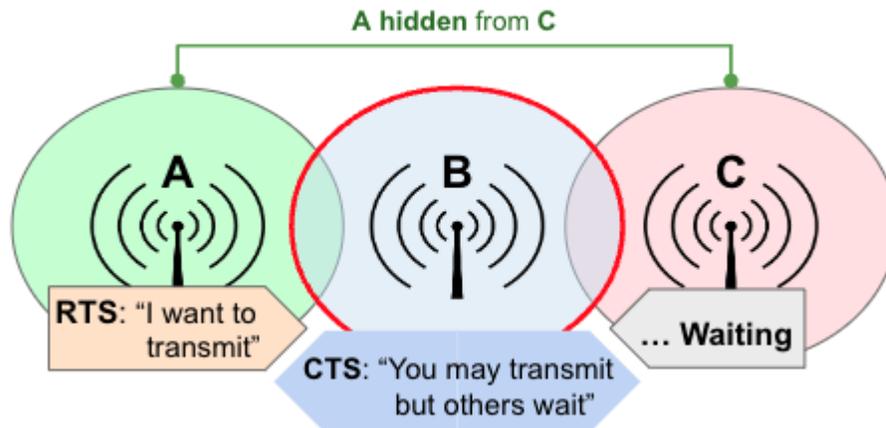
Several tools are available for testing network performance such as *ping* to measure latency, *traceroute* to identify how traffic is being routed, and *iperf3* to estimate network throughput. Periodic

measurements along with user perceptions can be helpful in determining whether channel separation would be of benefit. It is an expected by-product of how wireless networks normally operate, but performance can be enhanced by planning the assigned channels for your mesh devices as described in the **Channel Plans** section below.

13.1.2 Hidden and Exposed Nodes

Hidden Nodes

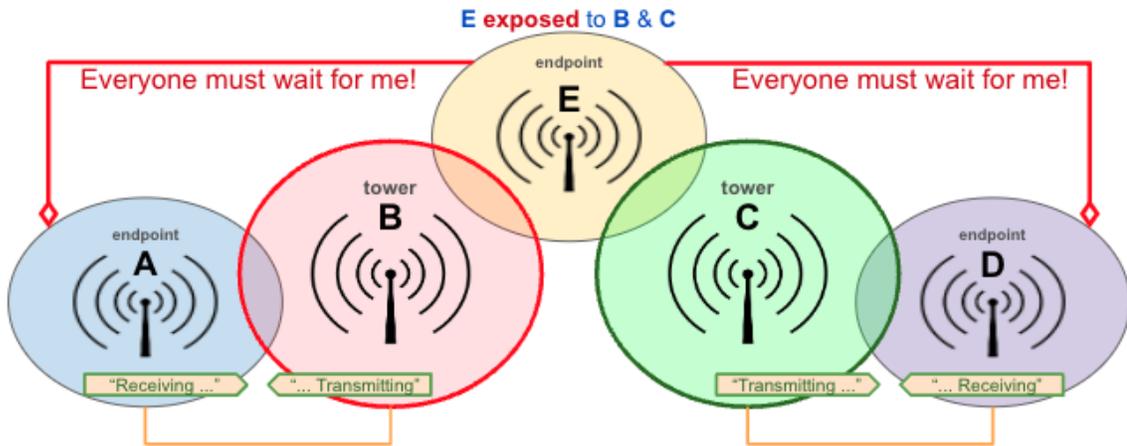
In any wireless network there will be nodes that are not within radio range of each other but which share the same channel. In the **Hidden Node** example below, node **A** can reach node **B** but cannot reach node **C**. Since **A** and **C** are hidden from each other, they may try to transmit on the shared channel at the same time without knowing it. Because of their relative locations and any associated network delays, each node may think it has a clear channel for transmitting.



Request to Send / Clear to Send (RTS/CTS) messages can be used by AREDN® nodes to minimize these issues. For example, node **A** broadcasts a short RTS message with a proposed timeslot/duration for transmitting its data stream. Node **B** receives that request and broadcasts a CTS for that time slot. Node **C** could not hear the original RTS but will hear the CTS message and defer its transmissions during that time slot.

Exposed Nodes

In the **Exposed Node** example below, Endpoint **A** and tower **B** can communicate with each other at the same time that tower **C** can communicate with endpoint **D**. However, if endpoint **E** is exposed to *both* of the towers, then the tower nodes will detect that the channel is not clear and will not be able to communicate when the exposed node is transmitting. This increases the network wait time which impacts overall throughput.



Try to eliminate the exposed node problem by placing them onto different bands or channels along with the nodes you want them to communicate with. Since nodes using directional antennas are nearly invisible to others not positioned in the antenna's beam, directional antennas should be used with care when sharing a channel so that exposed nodes are not created unintentionally. If you have exposed nodes that are causing throughput degradation, segment each group of nodes by putting them on different bands or channels.

13.1.3 Route Flapping

This is another issue that can lead to performance problems on a network. You may have parallel paths between two *Remote Nodes*, and these paths have similar ETX values which indicates that the cost of using either route is comparable. These two paths may appear to be functioning well most of the time.

However, when a bandwidth-intensive application such as video conferencing begins sending traffic across one of the paths, you may notice that link getting bogged down and the ETX will drop below that of the other path. At this point OLSR switches to the alternate path which now has a lower cost. The video stream then bogs down its new path, which lowers the ETX, and OLSR switches back to the original link whose ETX is better again. This situation may continue indefinitely, with neither path being able to deliver the traffic adequately.

This issue can happen on multi-hop links with similar ETX which seem to work fine until they are loaded with traffic. Then packet loss begins to occur, connections time out, and neither path is reliable during that cycle. One solution might be to improve the multi-hop link cost by increasing the signal quality of the links along one of the paths. Conversely, you could also turn down the power on the alternate path to increase its cost. If bandwidth-intensive traffic must be passed between two remote endpoints, the best approach would be to design a more robust path between those two endpoints to meet that need.

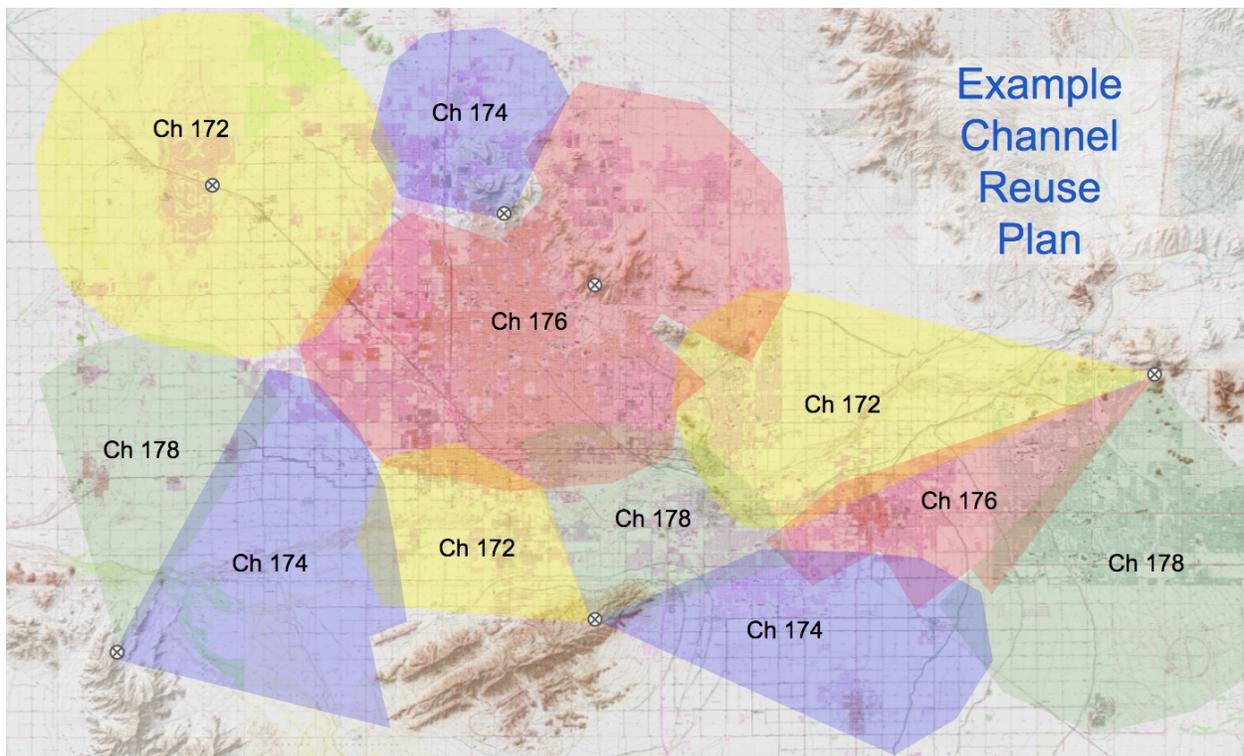
Another case is when there is one poor quality link over which all traffic must be routed. The handshaking and data retransmissions may cause all the other nodes to wait. The entire network

can be impacted by one low quality path which becomes a bottleneck. If at all possible you should increase the signal quality of that vital link or install a better link as an alternate path.

13.2 Channel Plans and Frequency Coordination

You may experience poor network performance if there are too many nodes using the same band and channel. Here is a simple example to illustrate the issue: a three-hop path from QTH1 to Tower1 to Tower2 to QTH2. If all links are using the same channel, then only one node at a time can send data. This instantly cuts the throughput by one-third or more and increases latency with protocol overhead. To improve performance you could configure each link to use a different channel, allowing simultaneous transmissions. For example, the collocated tower nodes could be DtD linked via Ethernet, with QTH1 and Tower1 using 5 GHz channel 172 while QTH2 and Tower2 use channel 176. Before this channel plan is implemented it might be possible to have one HD video stream and one VoIP call with frequent dropouts. After the channel plan is implemented it should be possible to have three HD video streams and several VoIP calls simultaneously with few dropouts.

Depending on the frequency band you are using, there are varying options available for assigning non-overlapping channels to your mesh devices. For example, in the 5.8 GHz band using even-numbered 10 MHz channels, there are 25 non-overlapping channels. Ideally, RF coverage zones (sometimes called “cells”) should use different channels. Overlapping cell coverage can provide broader connectivity, but the overlapping coverage zones should not use overlapping RF frequencies.



The example coverage map shows that four different channels have been assigned to achieve broad coverage by segmenting specific areas into zones to reduce co-channel contention. It should be noted that even a channel reuse plan such as this may not eliminate all instances of contention. For example, if a node is at the outer edges of a coverage zone or is elevated well above ground level, its transmissions may propagate into a distant cell using the same channel. The radios in the other cell will defer if they hear the original node's transmissions, even though they originate in a different cell. Some degree of experimentation may be required in order to minimize contention and maximize network throughput.

13.3 Collocated Nodes

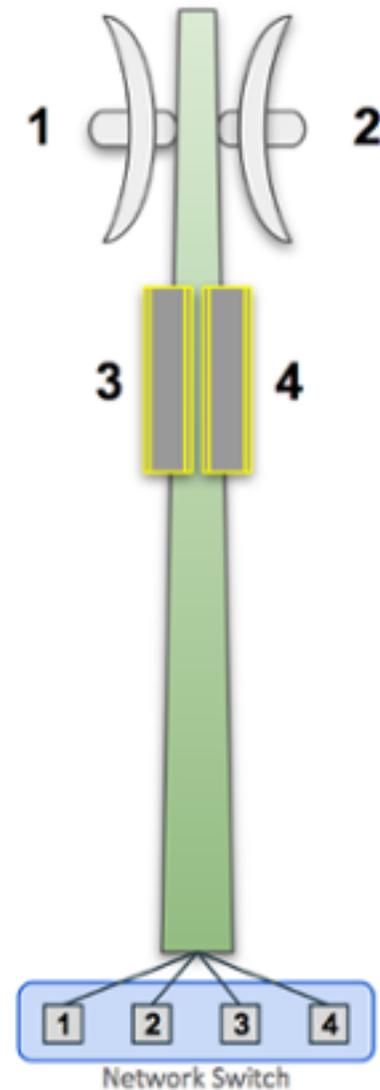
At some sites there may be several devices mounted on the same building or structure. This photo shows many nodes collocated on a mountaintop.



Network performance degradation can occur if these nodes share an RF band and channel. For example, when two sector antennas are collocated and share the same channel, the network throughput for that site will be reduced by half or more. If you have collocated nodes then it makes sense to allow the devices to pass traffic over their Ethernet interface (as described below) rather than forcing them to use their radio channel.

13.3.1 Device to Device (DtD) Linking

In its most basic configuration for two collocated nodes, an Ethernet cable is connected between the PoE *LAN* port of each device. OLSR will assign a very low “link cost” (0.1) to the DtD connection and automatically route traffic between the nodes over Ethernet rather than causing the RF channel to become busy.

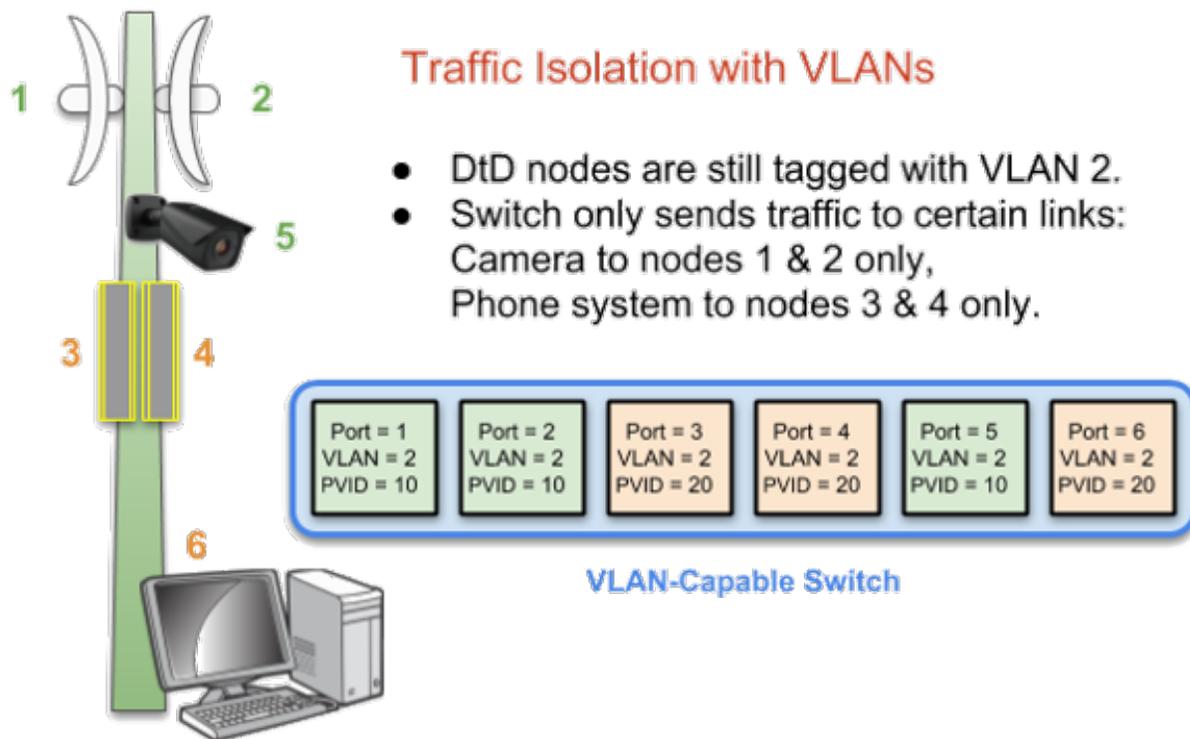


One added benefit of DtD linking is that you can link nodes which are operating on different bands and channels. Nodes that are using *Channel Separation* to segment traffic can still pass data at high speeds through their DtD link and be members of a single network. At a tower site like the one shown here, you could link 2.4 GHz and 5.8 GHz nodes to the same network. In fact, at a busy site like this it is best practice to use DtD linking, because otherwise RF channel contention could make the network unusable.

Ideally you should configure your collocated nodes to use different bands and channels, then set up DtD links between the nodes to ensure that traffic is routed efficiently without generating RF

contention or delays. OLSR will always choose the DtD path first when passing traffic between linked nodes. Each AREDN® node recognizes incoming packets tagged with VLAN (Virtual Local Area Network) 2 as DtD traffic. In the simple example shown here, the switch will share all traffic across all ports and every node will receive it on its DtD link.

If you want to partition traffic even further, you can configure VLANs on a managed switch to isolate port traffic so that only the nodes which should receive specific traffic will see it. For example, you may have a video surveillance system (5) or a VoIP (Voice over IP) PBX system (6), and traffic from those devices should only be passed to a specific set of links as shown in the diagram below. The port-based VLANs will ensure that traffic is controlled and routed, rather than being broadcast across every link.



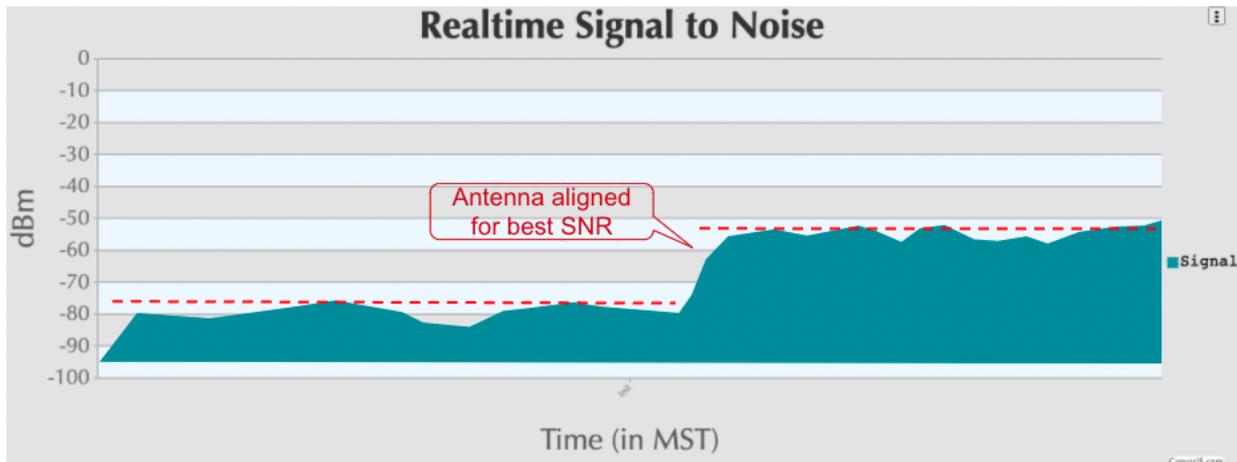
13.3.2 Antenna Polarization

Most of the latest AREDN® devices use dual polarity antennas and MIMO features in the radios that exploit multipath propagation. However, if you are using single polarity antennas with “single chain” radios, another way to achieve signal separation for collocated devices is to orient the site’s antennas so that one is vertically polarized and the other is horizontally polarized. This can result in a signal separation of up to 20 dB. Because of the predominance of vertical polarization in commercial WiFi devices, single chain AREDN® nodes may achieve slightly better performance

using horizontal polarization with clear line of sight. You can test both polarizations to see which one yields better performance dealing with the man-made noise in your specific environment. Note that the antennas on both sides of a radio link must be oriented the same way.

13.3.3 Aligning Linked Nodes

The AREDN® web interface provides information that is helpful when aligning two nodes that are being installed to form a link. On the **Node Status** page, click the **Charts** button to view the *Realtime Signal to Noise* graph. Slowly turn and tilt your antenna, pausing to view the signal metrics. Once you see the best signal, as shown below, you can lock your antenna into position. If you want to focus on the antenna position without having to watch the SNR graph, you can also enable the *SNR Sound* feature and align the antenna to the highest pitch tone. Depending on the implementation, a Signal to Noise Ratio of 15 dB is adequate to pass data at speeds in the range of 5 to 20 MBPS (Megabits per second). See “Tips for Aiming Directional Antennas” in the **How-To Guides** section for additional information.



13.4 Channel Planning Tips

Network Scalability Tip

If there are two towers or cell coverage areas within range of each other, configure the nodes with different channels to avoid poor performance.

Based on the purpose for your network, try to create reliable paths to the locations where data is

needed. Use channel separation and DtD linking of collocated nodes to avoid RF channel contention.

- If you need broad local coverage for a high profile area you can install sector antennas on a tower site: for example, three panels with 120 degree beam width each. DtD link the sectors at the tower site, and use different channels for each sector to avoid channel contention.
- Consider putting each local coverage area on its own channel to minimize the interaction between zones. Be sure to allow adequate RF separation between zones where channels are being reused.
- If you are installing long distance point-to-point links to connect network islands, be sure to use a separate band or channel for the backbone link. This type of link has a single purpose: to carry as much data as quickly as possible from one end to the other. Eliminate any type of channel contention so that these links can achieve high throughput.
- Remember that a multi-hop path through the network must have good signal quality on each leg of the journey. You cannot expect adequate performance through a series of poor quality links. For example, if you traverse three links having LQ (Link Quality) metrics of 65%, 45%, and 58%, your aggregate LQ will be 17% which is unusable. Ideally the aggregate LQ should be at least 80% to have a link that supports the applications and services you require.

[Link: AREDN Webpage](#)

NETWORK MODELING

As you design your AREDN® network it is often helpful to estimate ahead of time whether a node or link might accomplish your goals for the network. One way to get this information is to use computer modeling programs that predict the performance of RF devices. There are many types of computerized tools that you can use, ranging from relatively expensive commercial software to freely available open source programs. You should select and become familiar with the tool that best fits your aptitude, experience, and budget.

In this section some free tools will be used to illustrate how to determine your network's available paths and overall coverage. Keep in mind that a computer modeling tool only provides a prediction and does not guarantee that two sites will be able to communicate when actually deployed.

14.1 Creating a Path Profile

Path profiles are very helpful for determining whether a link between two nodes will have clear line of sight and acceptable signal levels. In order to create a path profile you will need to have the following information for both of your node endpoints:

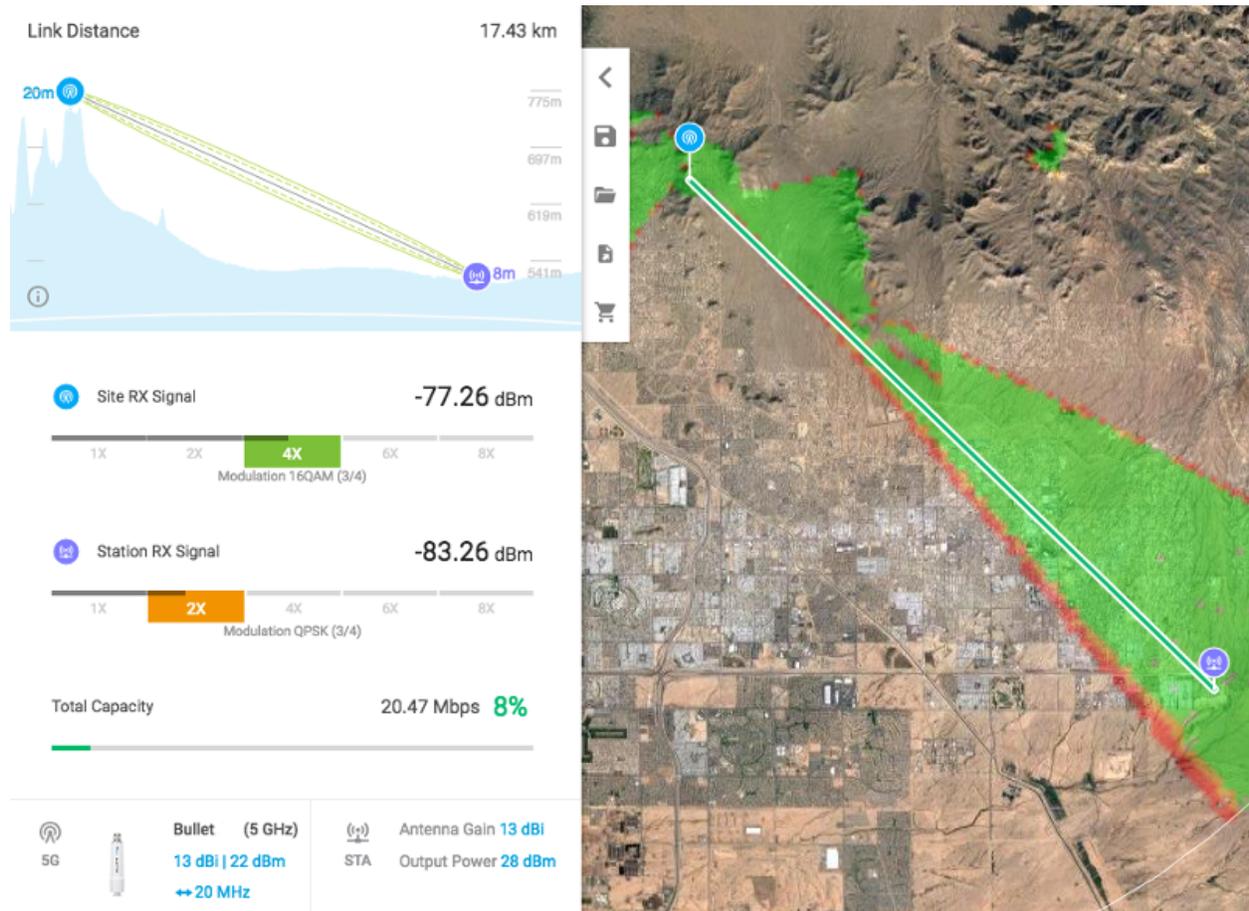
- Latitude and Longitude
- Antenna AGL
- Frequency
- Transmit Power
- Line Loss
- Antenna Gain
- Receiver Sensitivity

Most computer modeling software will be able to estimate the link characteristics given this information.

14.1.1 Ubiquiti AirLink Tool

If you are using Ubiquiti radios there is a free modeling tool available on the Ubiquiti website (<http://link.ubnt.com>). This tool will ask you to locate your node endpoints by clicking on a map display. It allows you to select the radio frequency and model from a dropdown list, as well as having you specify the antenna heights, antenna gain, and transmit power. With this information it will calculate and display the coverage area and the link quality.

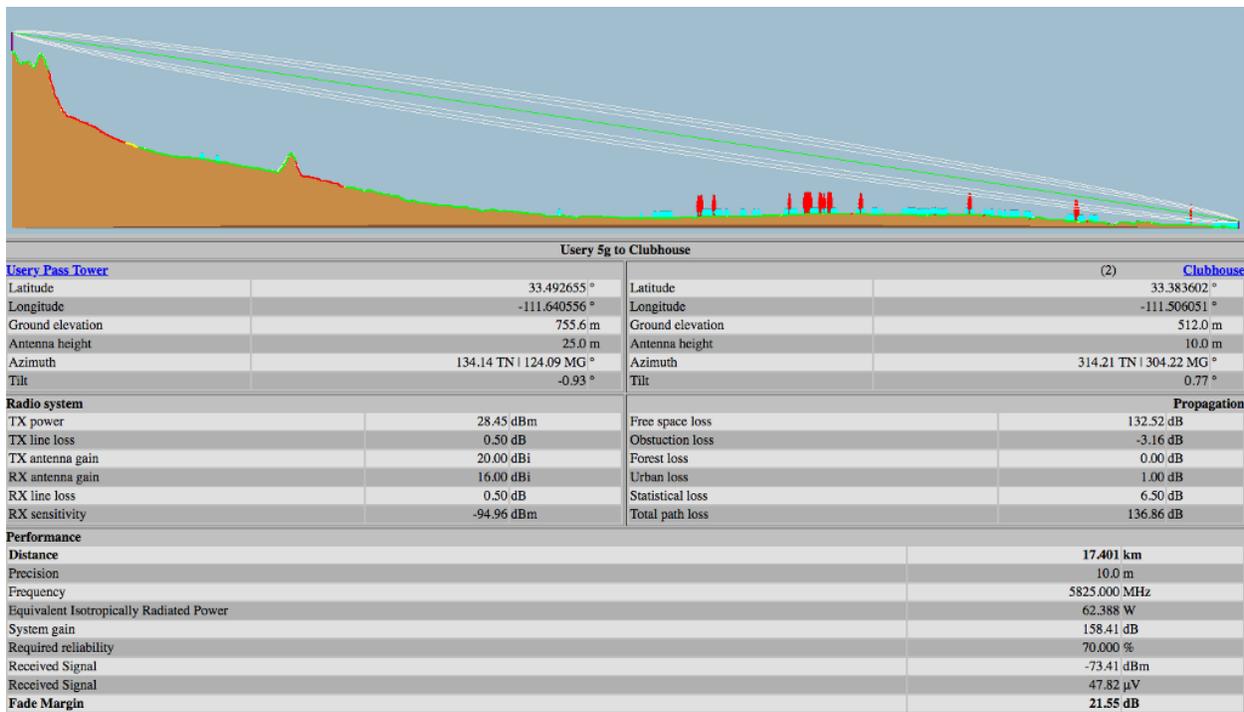
The path profile is color coded to indicate whether the link quality is adequate. It displays the link distance, line of sight, as well as the Fresnel Zone and 60% clearance area. It also estimates the signal levels at each endpoint and the predicted throughput for the link. An example *AirLink* path profile is shown below.



14.1.2 VE2DBE's Radio Mobile Tool

Whether or not you are using Ubiquiti devices, you can create detailed path profiles using VE2DBE's *Radio Mobile* software. This program is available for download, but it is very easy to use the web-based version: <http://www.ve2dbe.com/rmonline.html>

With *Radio Mobile* you must first create a *Site* for each of your endpoints. Then you can select the endpoints from your *Site* dropdown to generate a path profile between any of the listed locations. Once you enter the radio and antenna information in the link display, *Radio Mobile* will create your path profile. There are several metrics displayed here which may not be available in the Ubiquiti tool, including free space path loss, obstruction loss, forest loss, urban loss, and fade margin. This additional information may help you determine why a path is not working, and it may assist you with choosing alternate sites for node locations. Typically a fade margin of 15 dB or greater is adequate for a usable link. An example *Radio Mobile* path profile is shown below.

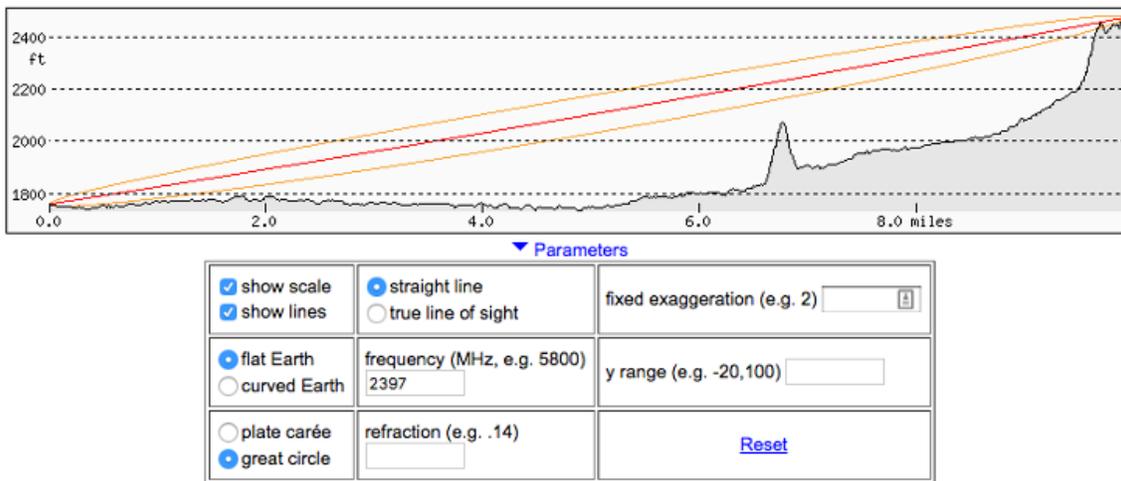


14.1.3 HeyWhatsThat Path Profiler

Another web-based tool will generate a path profile from points selected on a map. HeyWhatsThat Path Profiler is available here: <http://heywhatsthat.com/profiler.html>

Simply click on the map at the bottom of the webpage to add an endpoint for each side of your link. Once an endpoint has been added, it can be moved by clicking and holding the endpoint while dragging it to the new location on the map. After adding your endpoints you will see the path profile displayed at the top of the webpage. You can click the *Parameters* link under the path display to specify additional items for the path calculation. If you specify the frequency then the Fresnel zone for the path will be added to the display.

HeyWhatsThat Path Profiler



14.1.4 Radio Fresnel Tool

This web-based tool will generate a KML file which can be viewed as a 3D path profile using *Google Earth* software. Radio Fresnel is available here: <http://www.radiofresnel.com>

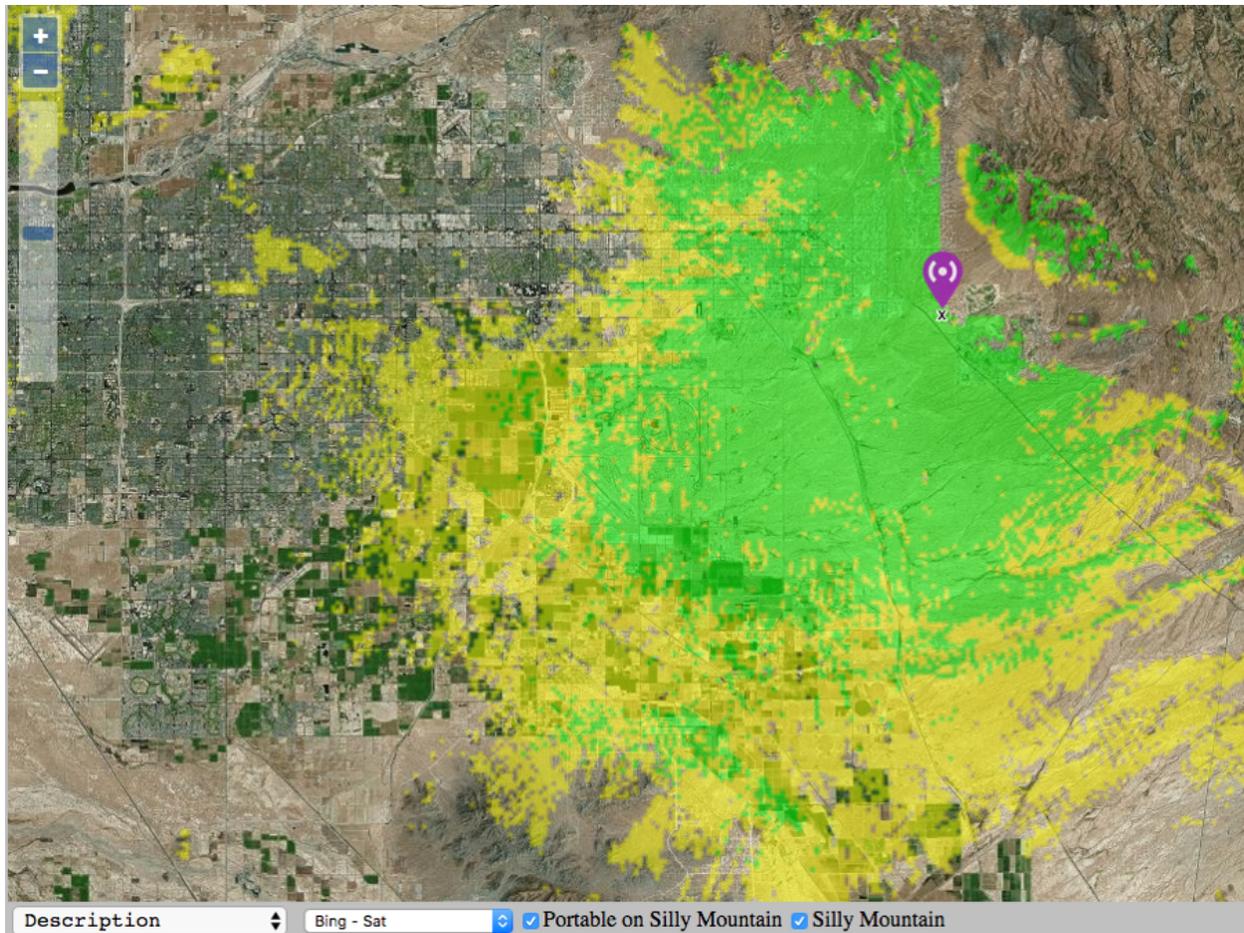
Simply enter the required site information into the online form and click the *Get KML* button at the bottom of the webpage. There is a sample KML file as well as a video tutorial for how to use the tool.



14.2 Determining Node or Network Coverage

In many cases it would be helpful to know ahead of time what area could potentially be covered with the signal generated by a particular node. Creating a coverage plot will show the predicted coverage on any of several types of base map.

An example *Radio Mobile* coverage plot is shown below. After entering the site, radio, and antenna characteristics the software produces a color coded map that predicts the areas of best, marginal, or no signal. One useful feature of *Radio Mobile* allows you to overlay several site coverage plots onto a single map so you can see the extent of coverage provided by multiple nodes in your network. Coverage maps such as these can show you the areas of adequate signal, as well as the “holes” which you may need to fill if you require more comprehensive coverage.



Link: [AREDN Webpage](#)

AREDN® SERVICES OVERVIEW

As mentioned in the AREDN® overview, the purpose of an amateur radio emergency data network is to provide typical Internet or intranet programs to people who need to communicate across a wide area during an emergency or community event. An AREDN® network provides the transport mechanism for the types of programs people typically use today to communicate with each other in the normal course of their business and social interactions. This may include keyboard-to-keyboard chat, email messages with images and attachments, file transfer, collaborative document sharing, VoIP phone service, video conferencing, GPS (Global Positioning System) tracking, surveillance camera streaming, computer aided dispatch, deployed resource management, weather station reporting, sensor monitoring and control, repeater linking, and many other services.

The purpose for this section of the AREDN® documentation is to identify examples of services that might be useful for communication across a mesh network. None of these programs are directly supported by the AREDN® development team. Almost any program that can operate on a peer-to-peer TCP/IP network is a candidate for AREDN® networking, but you should carefully select and test your services to ensure they will work within the following guidelines.

- An important consideration for selecting programs is to understand the impact each service will have on the performance and reliability of the network during the times when digital communication is required. As a best practice, choose programs which require the least amount of computing and network resources in order to operate successfully.

Note: The consideration above is especially important if you are deploying a service which regularly queries other nodes across the network. For example, if you deploy a network management system which polls metrics from remote mesh nodes, you need to carefully consider how many metrics you poll and how often you request them. Realize that polling dozens of metrics from each node every few seconds is likely to degrade mesh performance. Be sure to let node owners know what you are planning to do and get their permission/agreement for your polling schedule.

- It is equally important to choose data services that meet the criteria defined in FCC Part 97 regulations for amateur radio services. Try to avoid programs that use encryption or proprietary compression algorithms, which may be interpreted as “encoding messages for the purpose of obscuring their meaning” (FCC Part 97.113-a-4).

- As a general rule services should be run on separate LAN-connected computers rather than on the AREDN® nodes themselves. Node devices have very limited resources which should be conserved for node operation rather than for running extra programs. Try to select external computers that have low power requirements, since many AREDN® deployments are off-grid and without any external network access. Many operators use [Raspberry Pi](#) computers which are small, easy to transport, and require minimal DC power for operation.

When choosing programs to use as AREDN® services you will probably find that there is more than one way to accomplish your goals. It is crucial to clearly understand the types of communication that meet the requirements of your mission, and then you will be able to select the best programs for the job. Always try to use a program that will cause the least performance impact to your network.

Most TCP/IP programs are designed to use the [Client-Server](#) model, where one or more client programs communicate through a central server or servers distributed hierarchically. These types of programs can operate on a mesh network as long as the server is reachable or readily accessible by the nodes that need to use them.

As a general rule for mesh networks, simpler is better. The more complicated and automated you make your service design, the more network and computing resources will be required to operate the system. It is always best to conserve mesh networking resources wherever possible.

Several programs have been designed to take advantage of multiple paths between nodes and multiple peer servers coexisting on a mesh network. There are fewer of these mesh-friendly programs, but they will be identified as they appear in the following sections.

The remaining parts of this guide will focus on examples of services that could be offered on your AREDN® network. Programs are grouped by type, and where possible the network impact of each program will be described in order for you to understand the resources that may be required to use the program as a service on the mesh. Remember that the mentioned programs are merely suggestions or examples of typical Internet-style TCP/IP applications which could be deployed on you network to meet the specific communication requirements of your mission.

[Link: AREDN Webpage](#)

CHAT PROGRAMS

Online chat software includes any program which transmits short text messages between the sender and receiver. These realtime keyboard-to-keyboard messages create an environment similar to a spoken conversation. A chat session may involve one-to-one communication or group meetings. These programs are valuable for quick question/answer interactions where immediate replies are important. Timestamped conversation history is typically saved for future reference.

Chat programs are one of the least network-intensive types of communication programs, so they are a good candidate as low impact services on a mesh network. Many chat programs also offer file sharing, which allows you to get two functions within a single program. The following list is not comprehensive or complete but represents a sample of the types of chat programs that might be available for you to use as services on your mesh network. Only programs with open source licenses were included in this list, although commercial chat software can also be used.

16.1 MeshChat

MeshChat has become the primary chat service for AREDN® networks because it was written by Trevor Paskett K7FPV specifically for mesh communication. Users access MeshChat via web browser, and the service can run on the mesh node itself or on a LAN-connected Debian or Raspberry Pi computer. After logging in by entering a call sign, you can send a message by typing into a text box and clicking the *Submit* button. The list of active users is displayed, and every message is visible to all participants on the chat service. Multiple *Zones* and *Channels* are supported for categorizing and filtering message traffic.

A copy of the message database is stored on every device where MeshChat is running. Nodes may have intermittent network connectivity, but as long as at least one node is available the MeshChat database remains intact. Once nodes come online they immediately sync by retrieving a full copy of the message database. If any new messages are found, they are appended to the local message database.

In addition to the keyboard-to-keyboard chat feature, MeshChat also allows files to be shared between nodes. Files may be uploaded from or downloaded to the user's computer using the web

interface. If MeshChat is running on a radio node then the file storage is very limited, but if running on an external LAN-connected computer the file storage is limited only by the size of the disk that is allocated for MeshChat files.

MeshChat *Action Scripts* also provide for functional extensions, such as sending messages to an SMS gateway for external distribution. It is also possible for action scripts to periodically save the message database for archive purposes or integration with external tools.

Although MeshChat is a commonly deployed service, it is a third party package which is not available in the AREDN® repositories. You can find additional information by visiting this link: [MeshChat at Trevor's Bench](#)

As originally designed, MeshChat uses the Perl programming language and is able to run either on an AREDN® node or on a LAN-connected Debian or Raspberry Pi computer. After the retirement of Perl on AREDN® nodes, there are now alternative MeshChat packages which use the Lua programming language for running on nodes. If you are running the original Perl version on an external computer, you can still use the new Lua API on your node to provide the computer with the list of MeshChat nodes. These packages are available at the following links:

- [Latest Lua version of Meshchat at the new package maintainer's repository](#)
- [Older Lua version of Meshchat for AREDN 3.22.6.0 \(no longer maintained\)](#)
- [Original Perl version of Meshchat for AREDN 3.22.1.0 or for running Meshchat on a Raspberry or Debian computer \(no longer maintained\)](#)

CHAT FILES STATUS LOGOUT

Mesh Chat v1.0

Zone: MeshChat Node: ai6bx-2-chatpi
 Call Sign: KG6WX C Updated: 14 seconds ago

Send a Message

New Message

Enter message here

Channel:
Everything

Mesh Chat Users 1

Call Sign	Node	Last Seen
KG6WX C	ai6bx-2-chatpi	1/23/19 10:20 AM

Messages

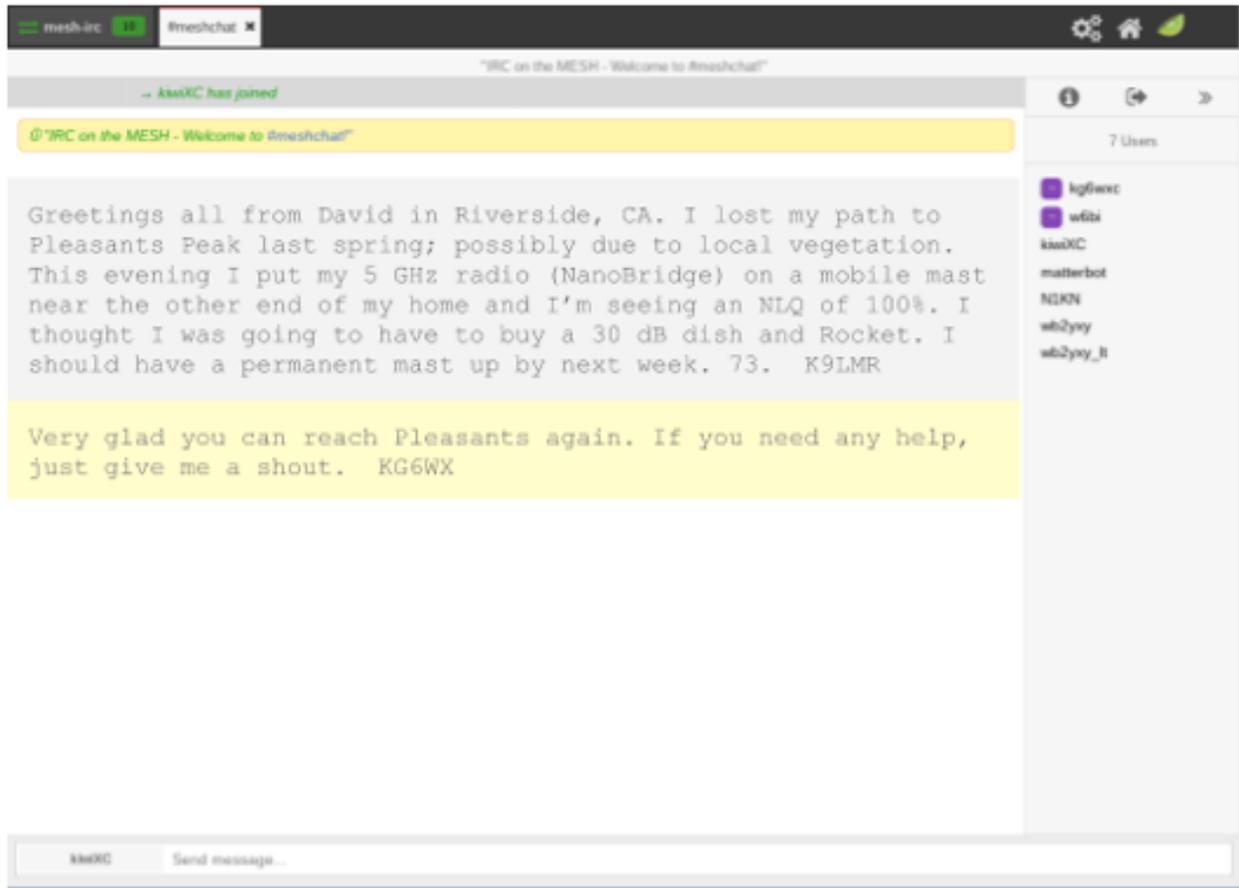
Enter search Everything

Time	Message	Call Sign	Channel	Node
1/16/19 7:13 PM	Greetings all from David in Riverside, CA. I lost my path to Pleasants Peak last spring; possibly due to local vegetation. This evening I put my 5 GHz radio (NanoBridge) on a mobile mast near the other end of my home and I'm seeing an NLQ of 100%. I thought I was going to have to buy a 30 db dish and a Rocket. I should have a permanent mast up by next week. 73.	K9LMR		ai6bx-2-chatpi

16.2 Internet Relay Chat

Several implementations of [Internet Relay Chat](#) are available, either as open source software or in proprietary versions. The Internet Relay Chat Daemon (IRCd) is a server program that listens for connections from IRC client programs and brokers the communication between the connected clients. With this client-server architecture, the IRC server must be available on a network link with sufficient bandwidth in order for the clients to function.

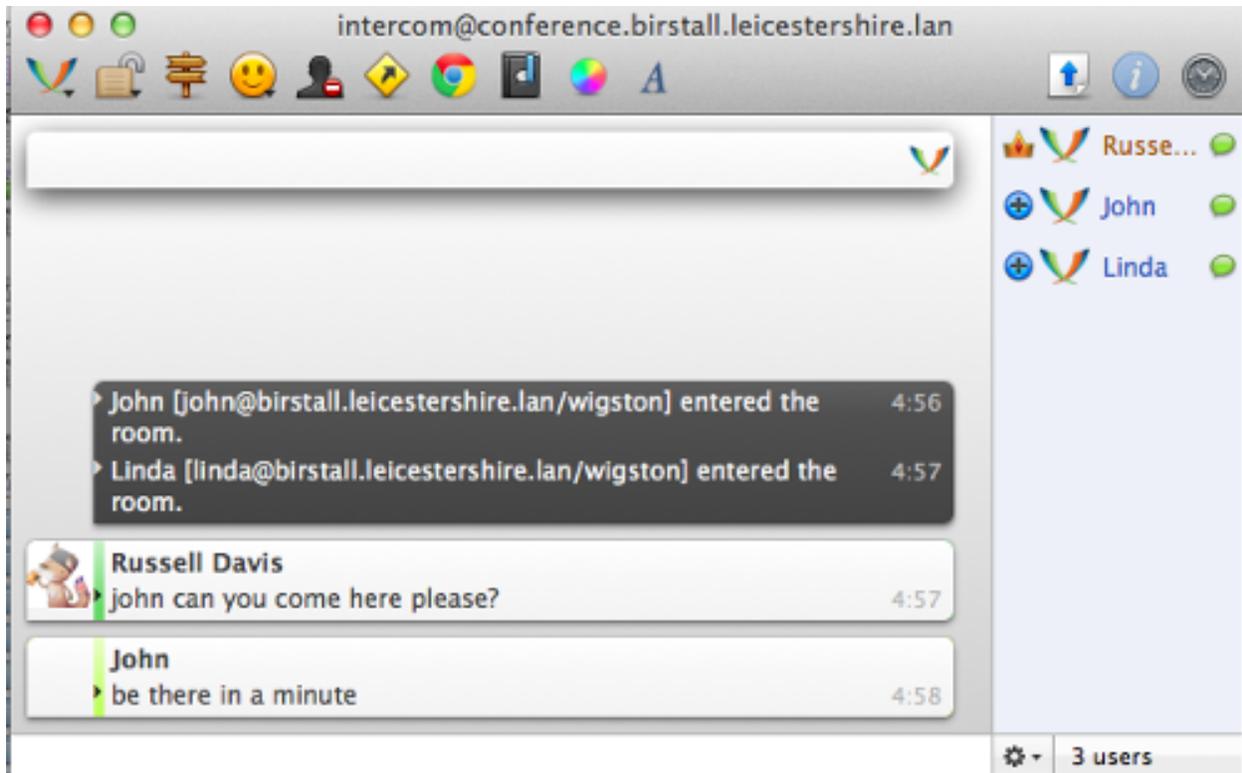
A wide variety of features and functions are available with these and similar chat programs, including various zones, channel types, and user roles. For additional information about IRC services, visit [IRC Clients](#)



16.3 Jabber/XMPP

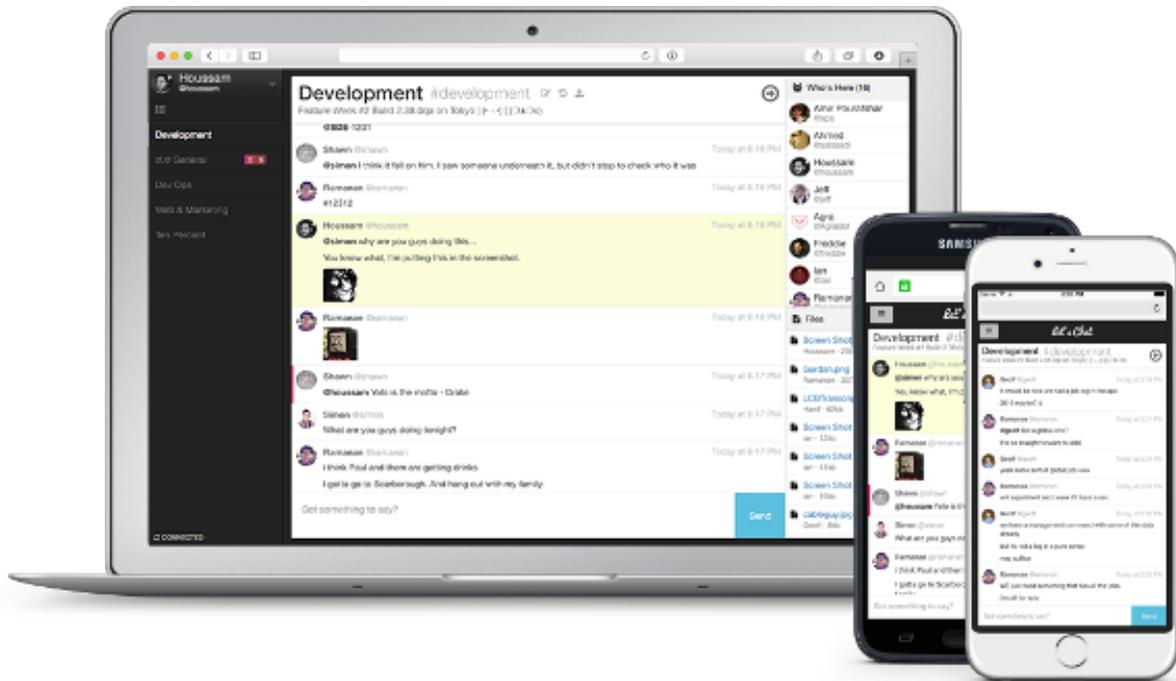
Originally known as Jabber, [XMPP](#) servers have been around for a long time but are fully compliant with modern messaging standards thanks to a large community of developers worldwide. These servers provide one-to-one messaging as well as group chat sessions. User lists have activity and presence indicators, and chat history can be archived for later use. There are dozens of feature modules available for XMPP servers which can extend the functionality as needed.

Two of the most popular XMPP servers are [eJabberd](#) and [Prosody](#), but there are many others. For additional information about these services, visit the following links: [eJabberd](#) and [Prosody](#)



16.4 Let's Chat

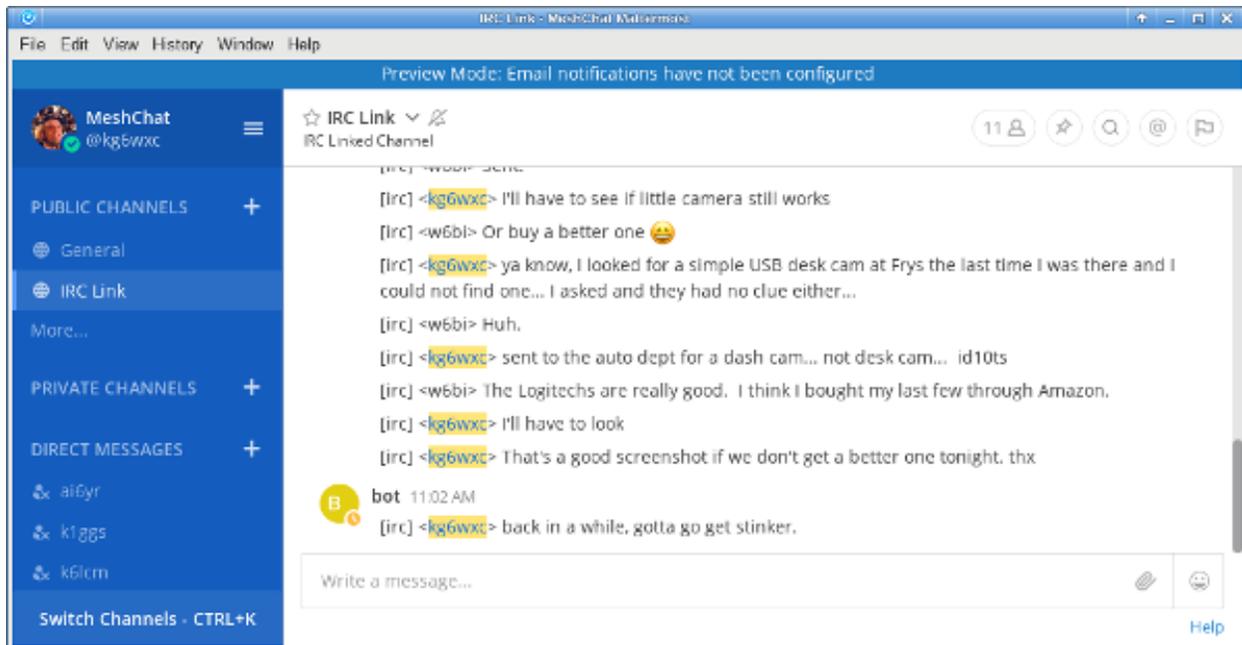
Let's Chat is an open source messaging service for small teams. It provides one-to-one communication between XMPP users as well as group messaging and @mentions in a variety of chat rooms. Searchable conversation history is available, in addition to text and image pasting, user activity notifications, and file uploads. User self-registration is configurable on the server. For additional information about Let's Chat, visit this link: [Let's Chat](#)



16.5 Mattermost

The *Mattermost Team Edition* is an open source platform that supports mobile and desktop messaging apps. It provides one-to-one and group messaging, file sharing, and message history with search capabilities. It is often described as an open source alternative to the commercial *Slack* communication tool.

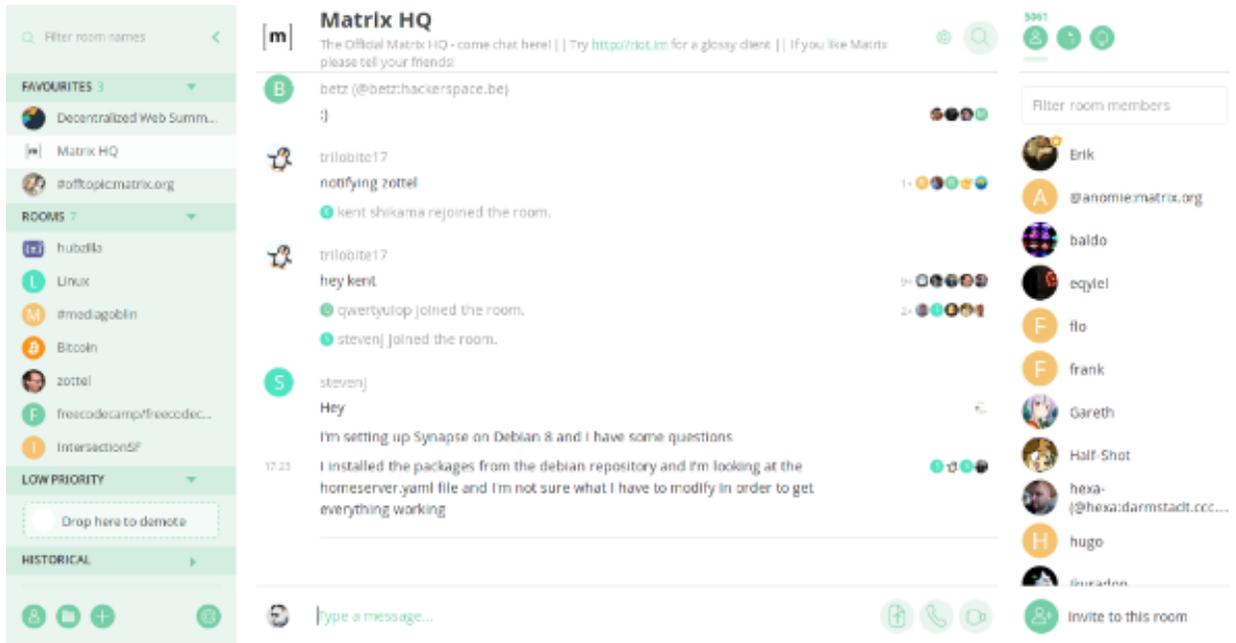
Mattermost supports @mentions, and channels are available for organizing conversations which can be topic-based, group-based, or event-based. Notifications indicate user presence and activity. File sharing is provided for PDF and text files, as well as audio, video, and image files. For additional information about Mattermost, visit this link: [Mattermost](#)



16.6 Matrix - Synapse

Synapse is the “homeserver” implementation of the *Matrix* communication platform. As with a traditional client-server architecture, every user runs a Matrix client that connects to a Synapse server which stores the personal chat history and user account information. However, these servers communicate with each other on the network, which creates a distributed content architecture that minimizes single points of failure.

Matrix services can provide one-to-one communication channels as well as group chats in a variety of rooms. User presence and typing notifications are supported, as well as chat history and read receipts. Although the Matrix platform is intended to provide end-to-end encryption, it can be run without cryptographic signing. Matrix can also integrate with IRC (Internet Relay Chat) services, as well as VoIP and video conferencing solutions via [WebRTC](#). For additional information about Matrix-Synapse, visit these links: [Matrix Home](#) and [Synapse](#)



16.7 Example Chat Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	Architecture	Network Load	Age	Platform	Effort
MeshChat	mesh aware	small	new	node/rpi	easy
IRCd server	client-server	small	old	lin/mac/rpi/win	medium
Jabber/XMPP	client-server	small	old	lin/mac/rpi/win	medium
Let's Chat	client-server	small	new	lin/mac/rpi/win	medium
Mattermost	client-server	medium	new	linux	expert
Matrix	distributed	medium	new	linux/mac	expert

Link: [AREDN Webpage](#)

EMAIL PROGRAMS

Email programs have become a communication standard for workers everywhere today. Email messages can include a wide range of information, from short chat-like interactions to lengthy and extensive text with complex document and image attachments. Whereas chat programs often assume that the sender and receiver are online at the same time, email programs use a [store and forward](#) approach to ensure message delivery even when users are not connected simultaneously.

Email operates on a client-server model. Users create or read their messages on some type of client program, although this software could be hosted on a network web server and accessed through a user's web browser rather than requiring a standalone email program to be installed on the client computer. Client programs typically access messages from the email server using either [Internet Message Access Protocol \(IMAP\)](#) or [Post Office Protocol \(POP\)](#). Client programs use [Simple Mail Transfer Protocol \(SMTP\)](#) to send messages to email servers, while the servers themselves use SMTP for both sending and receiving.

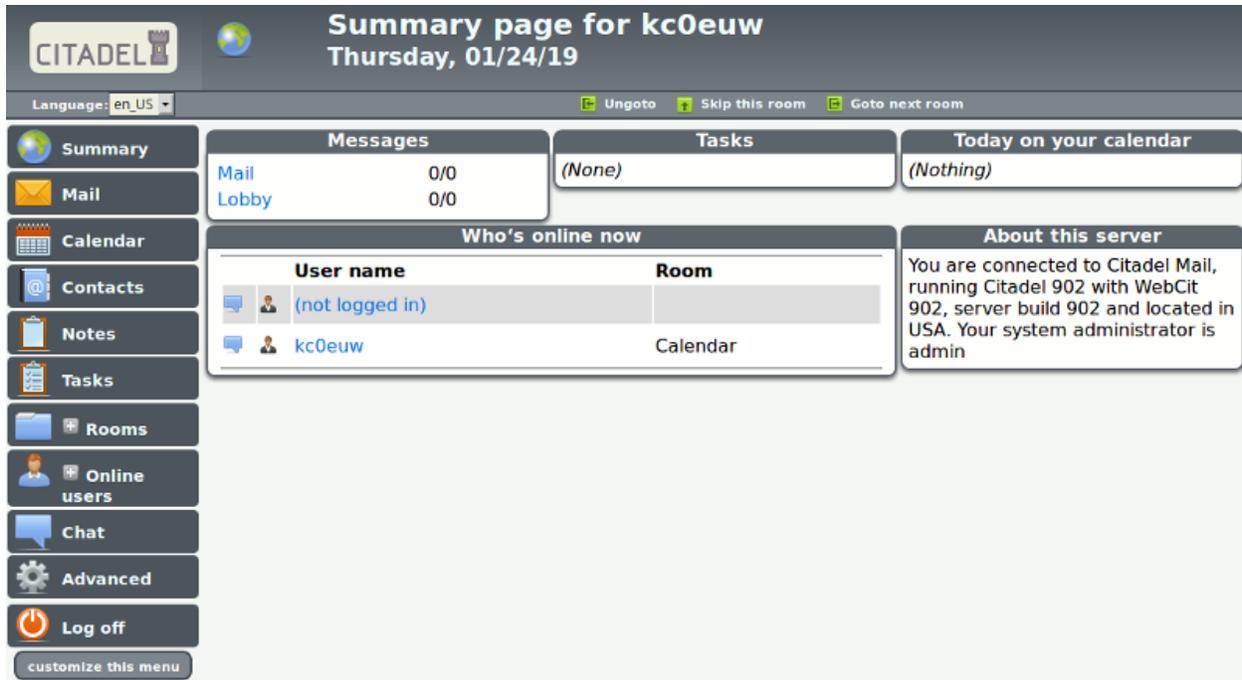
As with any client-server program, the email server must be reachable on a network segment with adequate bandwidth in order for the clients to exchange messages. If you have a choice, put your email server on one of your largest and most reliable network segments. Refer to this link for a comparison of email [Client Programs](#), and visit this link for a comparison of email [Server Programs](#). The following list is not comprehensive or complete but represents a sample of the types of software that may be available for you to use as services on your mesh network. With one exception, only programs with open source licenses were included in this list, although proprietary email software can also be used.

17.1 Citadel/UX

Not only does Citadel provide email, but it is also a full-featured *groupware* suite with chat rooms, calendars and scheduling, contact address book, file sharing, forum posting, and many other features. It contains built-in implementations of the following server protocols: IMAP, POP3, SMTP, XMPP, and ManageSieve. Citadel also provides user self-registration, which minimizes the administrative overhead of managing email addresses on the server.

Since a variety of features are bundled into a single application suite, Citadel is a less compli-

cated and more integrated way to implement several network services at once by installing a single package capable of running on a lightweight [Raspberry Pi](#) computer if necessary. Citadel's email services can be accessed using its browser-based webmail interface or from a separate email client program on a remote computer. For additional information about Citadel, visit this link: [Citadel](#)



The screenshot displays the Citadel webmail interface for room `kc0euw` on Thursday, 01/24/19. The interface includes a sidebar with navigation options: Summary, Mail, Calendar, Contacts, Notes, Tasks, Rooms, Online users, Chat, Advanced, and Log off. The main content area is divided into several sections:

- Messages:** Shows counts for Mail (0/0) and Lobby (0/0).
- Tasks:** Displays "(None)".
- Today on your calendar:** Displays "(Nothing)".
- Who's online now:** A table listing online users:

User name	Room
(not logged in)	
kc0euw	Calendar
- About this server:** Provides details: "You are connected to Citadel Mail, running Citadel 902 with WebCit 902, server build 902 and located in USA. Your system administrator is admin".

17.2 Open Source Email Server

In order to implement an open source email server you will need to install several individual software packages, each of which will process one or more of the required email protocols. This is slightly more complicated than implementing a single groupware package such as the *Citadel* program described in the previous section. Protocols and example packages are described in the following lists.

SMTP

In order to implement an email server you will need to select a software package to handle the Simple Mail Transfer Protocol. You can select one of the example open source packages from the list below, or you can implement another SMTP agent of your choice.

- [Sendmail](#) is the original legacy SMTP server that is still used today, although one of the newer programs below is often chosen for its ease of configuration and added security features.

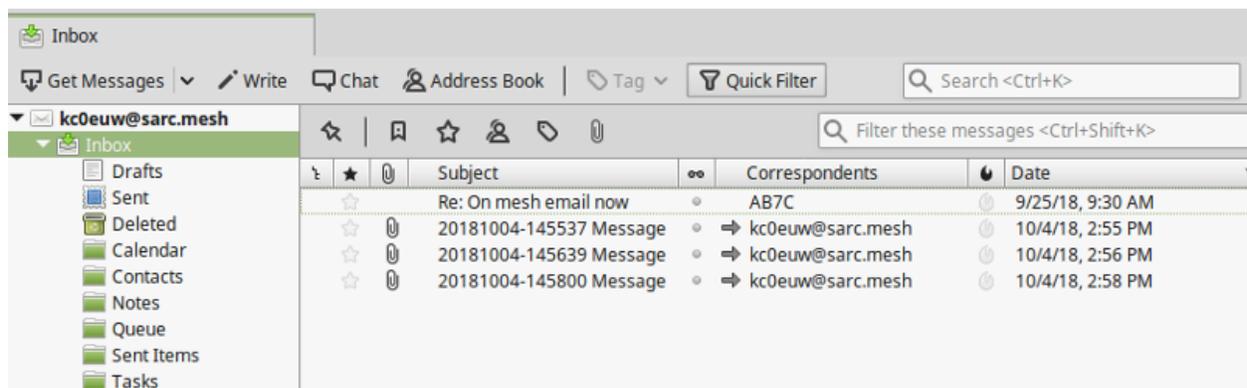
- [Exim](#) is the default SMTP server in Debian Linux, is well-documented, having many configurable features, and it runs from a single executable program.
- [Postfix](#) is the default SMTP server in Ubuntu Linux and MacOS, with many integration and security features, and it runs a series of parallelized programs for improved performance.

IMAP and POP3

In order for email clients to retrieve their messages you will need to select a software package to handle IMAP and POP3 communication. You can select the example open source package below or you can implement another IMAP/POP3 package of your choice.

- [Dovecot](#) is one of the most popular IMAP and POP3 servers for open source email systems, being found on more than 2/3 of the email servers across the Internet.

You will need to have detailed knowledge and skills when building your own open source email server, with the advantage of having complete control over everything on the system. There is some administrative overhead for creating and maintaining all user email accounts as well as handling other management tasks on your system. Using these open source software packages, it is possible to build a very robust email server that is capable of running on a small portable computer like a Raspberry Pi.

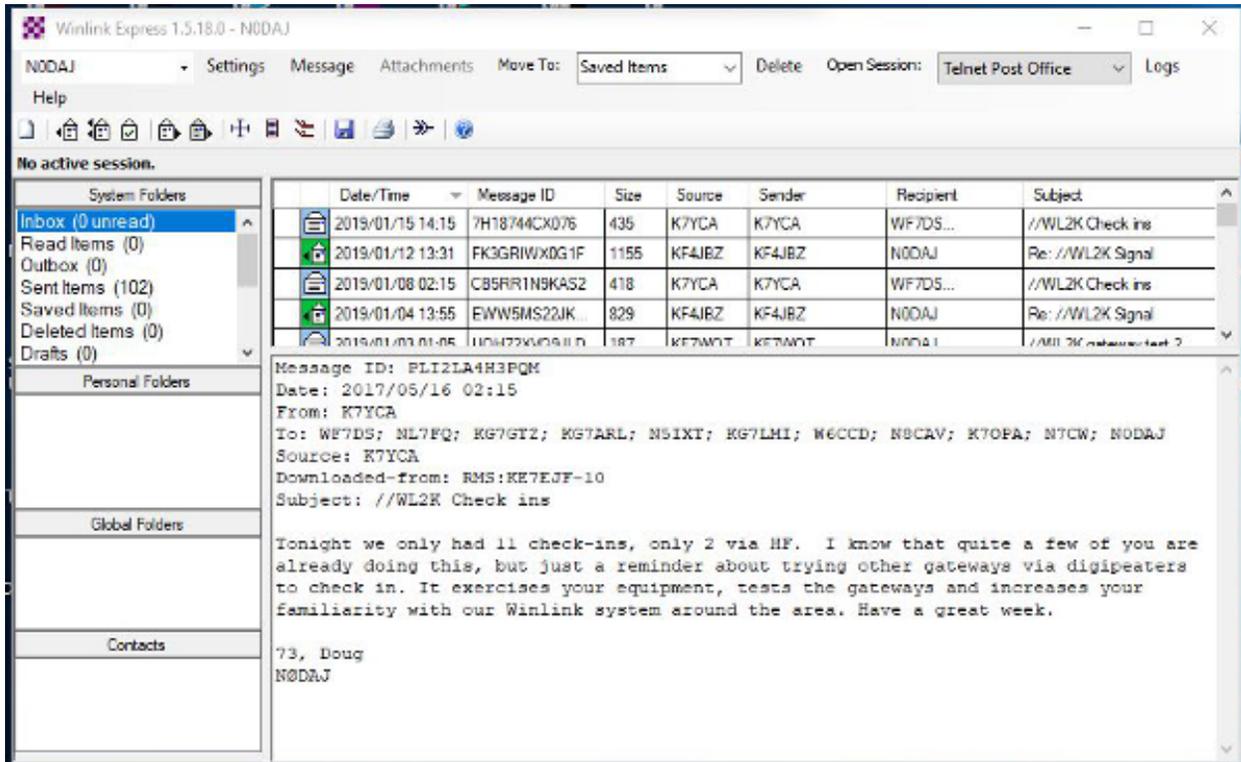


17.3 Using WinLink to Send Email

Although it is not typically used as a TCP/IP network application, many operators are already familiar with [WinLink 2000](#) for sending message traffic between WinLink computers across amateur radio frequencies. It is possible to configure *Winlink Express* and Telnet Post Office or Telnet P2P for sending email with attachments across a mesh network.

You will need a stable Microsoft Windows computer with plenty of memory to run this system (8GB recommended). The maximum attachment size is currently 5MB per message as compared

to the 100KB limitation on HF and Packet RMS stations. Refer to the information below for details about specific network settings and procedures for configuring Winlink over AREDN®. Additional information compiled by Orv Beach W6BI can be found in the [document linked here](#).



17.4 Example Email Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	Features	Network Load	Platform	Effort
Citadel	groupware, webmail	small	lin/mac/rpi	easy
Open Email	client-server	small	lin/mac/rpi	expert
WinLink	email, attachments	small	win (proprietary)	medium

Link: [AREDN Webpage](#)

FILE SHARING PROGRAMS

File sharing is a method of providing network users with access to digital content. One way to accomplish this is to *push* a copy of a file to users' computers, using either an email attachment or a file transfer program. Another approach is to create a central repository and allow users to *pull* files from this file share. Unless there is a special reason for pushing content, it is usually preferable to let users pull content as needed.

File transfer protocols themselves have minimal impact to network performance, but downloading a very large file across a mesh network could have a major performance impact. Transferring text files, and especially compressed text, should have minimal impact to the network, but a network could experience performance degradation while transferring files with lots of embedded formatting directives or images. High resolution audio files, image captures, or video recordings will also tax network resources when they are moving between nodes.

The following list is not comprehensive or complete but represents a sample of the types of programs that might be available to use for file sharing on your mesh network. Only programs with open source licenses were included in this list, although commercial software can also be used.

18.1 FTP Services

File Transfer Protocol (FTP) servers can be configured as file repositories from which users can copy digital content using FTP client programs. Some of the more common FTP server packages include **FileZilla Server**, **ProFTPD**, **Pure-FTPd**, and **vsftpd** (which is the default FTP server in many Linux distributions).

All of the most common web browsers allow content to be downloaded using FTP as shown below, although they may not support all protocol extensions. However, there are many **FTP client programs** with complete FTP support. FTP is a tried-and-true method for retrieving files from a central repository.

Index of ftp://n7qjk-host.local.mesh/

↑ [Up to higher level directory](#)

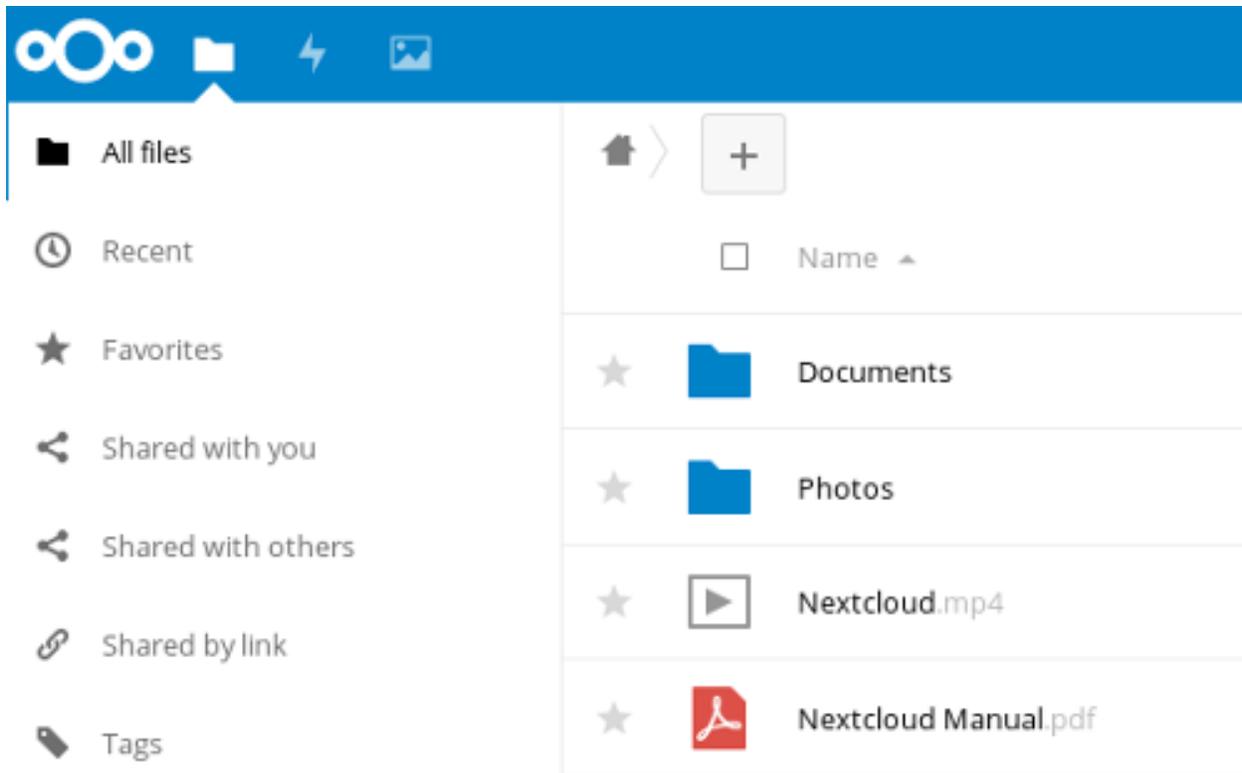
Name	Size	Last Modified	
File: Camg sample email mail setup for Icedove and Thunderbird - POP3.pdf	106 KB	6/3/18	12:00:00 AM MST
MeshChat		1/17/19	7:51:00 AM MST
Misc Files		9/24/17	12:00:00 AM MST
File: N7QJK FTP Welcome Msg.pdf	22 KB	9/24/17	12:00:00 AM MST
PDF Files		11/30/18	5:09:00 PM MST
RPi Files		5/13/18	12:00:00 AM MST
Simple Machine Forums		7/24/18	12:00:00 AM MST
Uploads		1/17/19	7:51:00 AM MST

18.2 Web Services

File sharing can be accomplished by hosting downloadable files on a web server. These files can be downloaded from within web browsers using [Hypertext Transfer Protocol \(HTTP\)](#) as well as other built-in file transfer protocols. Simply place files to be shared into the website directory structure and provide links to them on web pages.

There are also many web service packages that provide a robust file sharing interface similar to online cloud storage solutions. One example is [NextCloud](#), an open source file hosting suite with features similar to many of the Internet-based [cloud storage services](#).

Users login to NextCloud to see available content, and file sharing permissions can be set on a user or group basis. Files and folders can be uploaded, downloaded, moved, renamed, deleted, and previewed (depending on file type). Simple file version control is provided through auto-backup, and the *Details* sidebar lists past versions available for rollback. These and other similar software packages can provide a full-featured file sharing service when hosted on a web server.



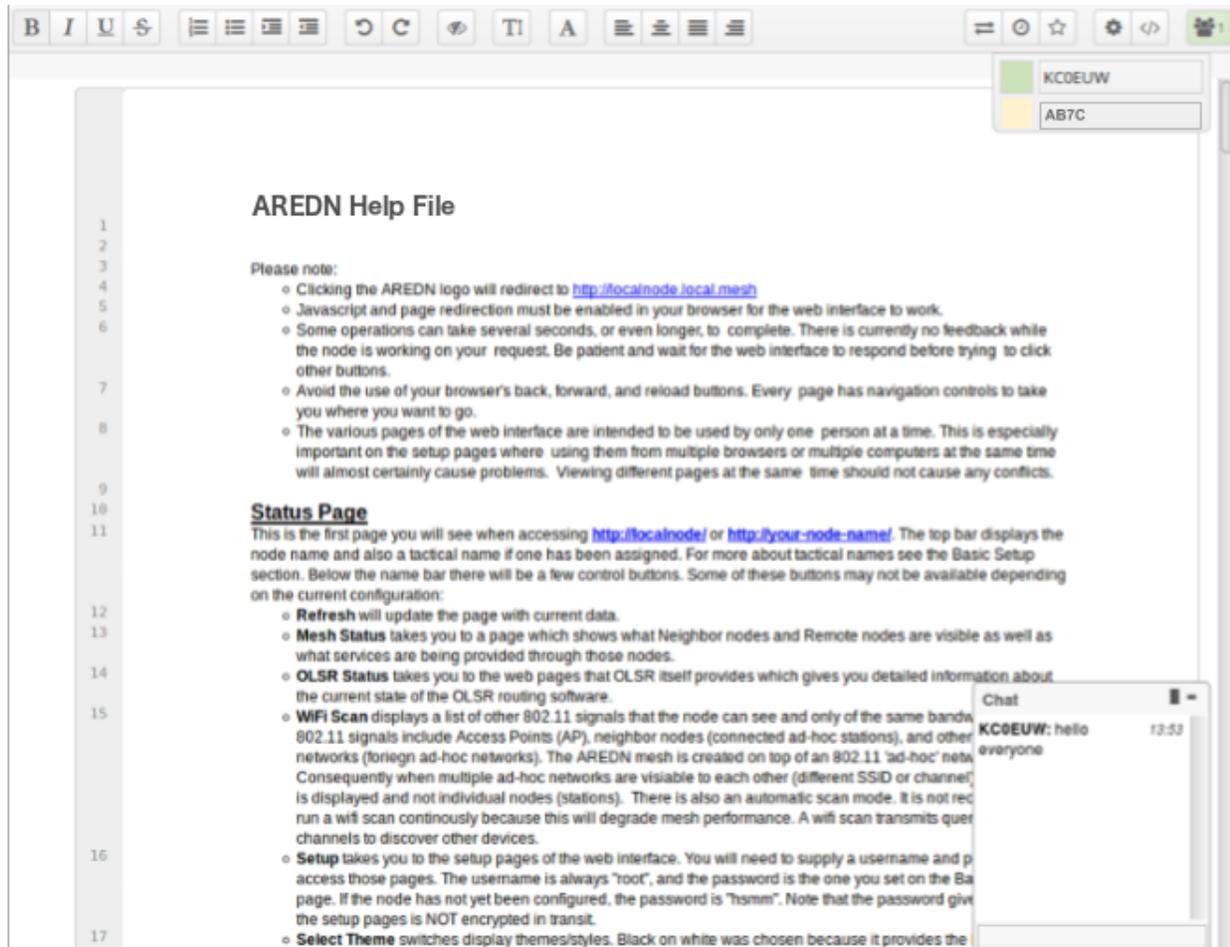
18.3 Collaborative Computing

Collaborative computing enables people to collaborate on documents in real time. Multiple users dispersed across a wide geographic area can be working simultaneously to create or modify a set of documents that are available to others over the network. With this type of collaborative model, documents no longer need be viewed as static but can become truly living projects.

One example package that facilitates collaborative document creation is [Etherpad Lite](#). Users access the Etherpad server through a web browser, so no client software is required on the users' computers. Anyone who connects to the service can create a new document or contribute to an existing document. Active users are displayed and have the ability to chat with each other in the messaging area. Changes to a document are periodically auto-saved, but users can force a checkpoint to capture the current state of a document. The "time slider" control allows users to view document revisions at any point in time throughout its history. Documents can also be downloaded in several formats (text, HTML, Open Document, Microsoft Word, or PDF).

[Collaborative document sharing](#) could be very helpful for a number of EmComm use cases, such as maintaining an accurate picture of deployed resources at various locations during an incident or event. Document version tracking makes it possible to scroll back and forth in history to see

the status of deployed resources at any given time, as well as to capture information and save it for wider distribution.



Link: [AREDN Webpage](#)

VOIP AUDIO/VIDEO CONFERENCING

The programs described in the previous sections can facilitate the sharing of detailed information across your mesh network. Some of them attempt to emulate a conversation, but nothing can replace an actual interactive discussion. Today people are accustomed to voice conversations, and since much of a message is communicated by non-verbal queues, having an audio-visual conversation can be even more effective. However, these communication advantages come at a cost. Multimedia programs will typically have a much greater impact on network performance than the programs mentioned previously.

The software described in this section can help you to provision services that enable both voice and video conferencing on your network. The phrase [Voice over IP \(VoIP\)](#) encompasses a collection of technologies capable of encoding and delivering realtime multimedia content across a digital network. When you have an established need for this type of communication, and if your mesh network is capable of supporting it, there are many reliable options for implementing VoIP and video conferencing.

The following list is not comprehensive or complete but represents a sample of the types of software that may be available for services on your mesh network. With one exception, programs having open source licenses were included in this list, although software with proprietary licenses can also be used. Dozens of VoIP programs have been available over the years, but the list of current open source projects in active development has dwindled over the past decade. Refer to this [link](#) for a comparison of [VoIP client and server software](#).

19.1 VoIP Server

Asterisk Server

[Asterisk](#) is one of the original *software* [Private Branch eXchange \(PBX\)](#) servers. It was first designed to run on Linux computers, but it is now available for MacOS and OpenWRT routers. It has been used to build large-scale telephony systems so it has many of the features of commercial and proprietary PBX systems, including voice mail, conference calling, interactive voice response (IVR) menus, and automatic call distribution.

Dozens of full-length books have been written about Asterisk, so it is widely documented.

It also serves as the underlying communication engine for several other software PBX packages. Asterisk is extremely robust tried-and-true IP-PBX software, but you will need specific knowledge and skills to implement it.



FreePBX Server

FreePBX is a web-based graphical user interface (GUI) for managing Asterisk. However, it is most commonly deployed as part of the integrated **FreePBX Distro**, which installs a complete Linux operating system with Asterisk, FreePBX, and software dependencies included.

All of the extensive features of Asterisk are available along with the benefit of having the FreePBX web interface to facilitate Asterisk management, making it much easier for users who are not telephony experts. Many mesh network operators who deploy VoIP have taken advantage of the *FreePBX Distro* when implementing their PBX services.



19.2 VoIP Endpoints

Once you have a VoIP PBX provisioned on your mesh network, you will need VoIP endpoints which can communicate through the server. Specialized [VoIP phone](#) hardware is available from several manufacturers which can provide communication endpoints on your network. It is also possible to use legacy analog phone hardware connected to the network using [Analog Telephone Adapters \(ATA\)](#). In addition to these options, there are pure software phones ([softphones](#)) that are supported on a variety of devices, such as the Linphone program described below.



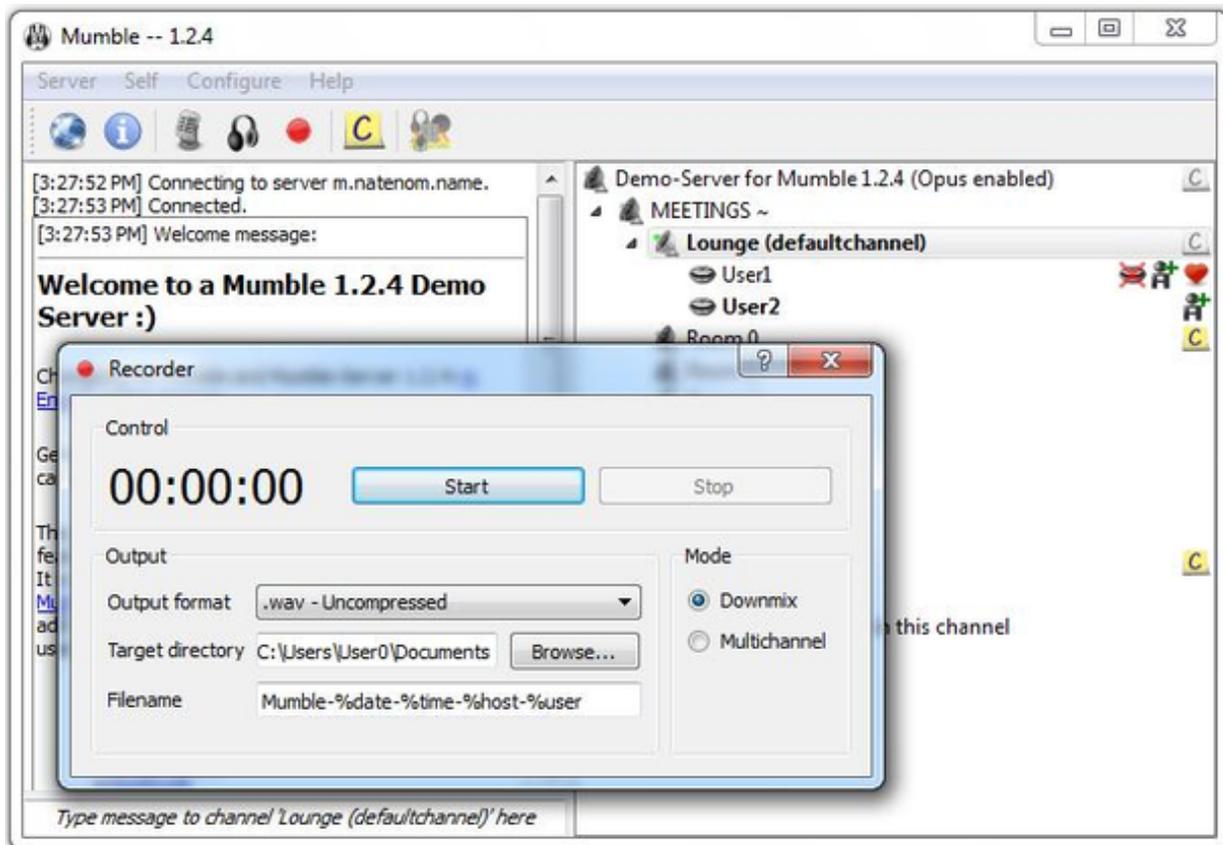
Linphone Softphone

Linphone is a software phone that is supported on Windows, Linux, MacOS, Raspberry Pi, iPhone, and Android. It can be used to place voice and video direct calls as well as calls through a VoIP PBX like those mentioned above. Users can transfer calls to other numbers, send chat messages, share pictures or files, and merge calls into a group conference. The softphone has the ability to manage contact lists, and call history is available for future reference.

Mumble

Mumble is a VoIP package that is available on Linux, MacOS, and Windows systems which support the Qt platform. Mobile apps are also available, such as *Mumblefy* for iPhone and *Plumble* for Android.

Hosting Mumble locally requires downloading the *Murmur* server, which is included as an option in the Mumble installer. The primary users of Mumble are Internet video gamers who want to communicate with each other during game play. However, it can also be used as a non-gaming voice communication service which does not require that an IP-PBX server exist on the network.

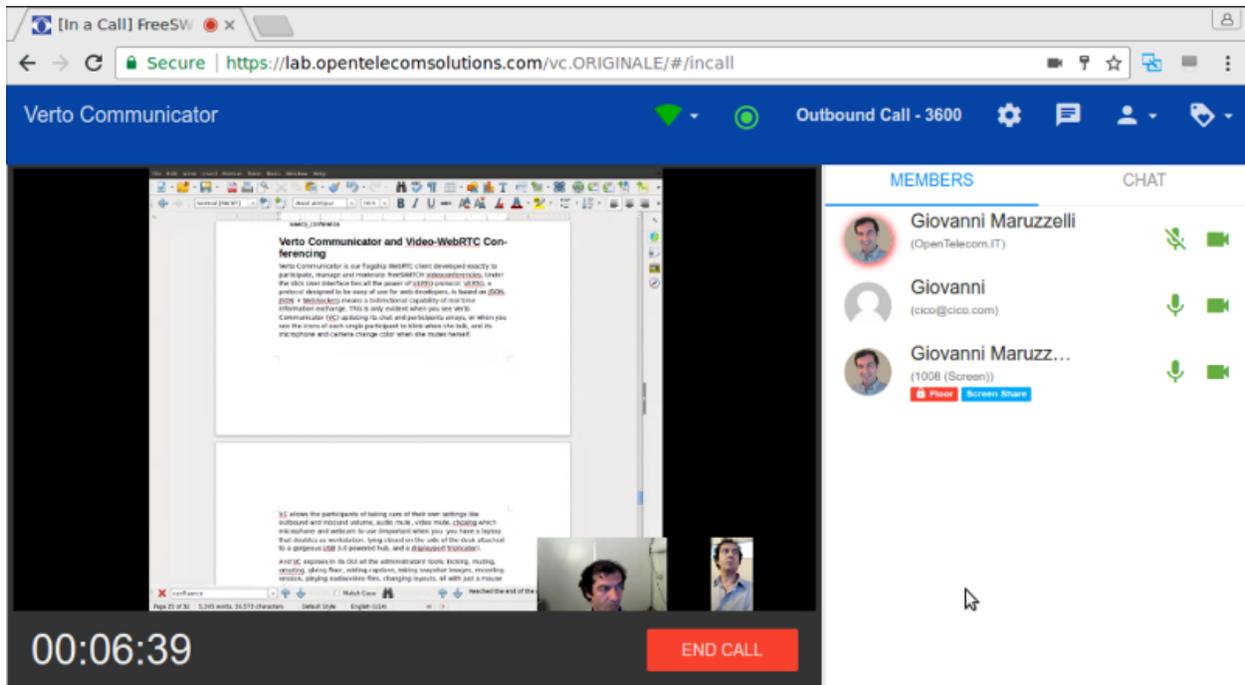


19.3 Video Conferencing Software

FreeSWITCH Server

FreeSWITCH is a recent communication platform that can be used to build voice PBX systems with voice response menus, video conferencing with chat messaging and screen sharing capabilities, and full **WebRTC** support. Its modular design makes it possible to install only what is required to meet your communication needs. Currently the FreeSWITCH package can be installed on Linux and Windows servers, and it can be compiled on MacOS computers if required.

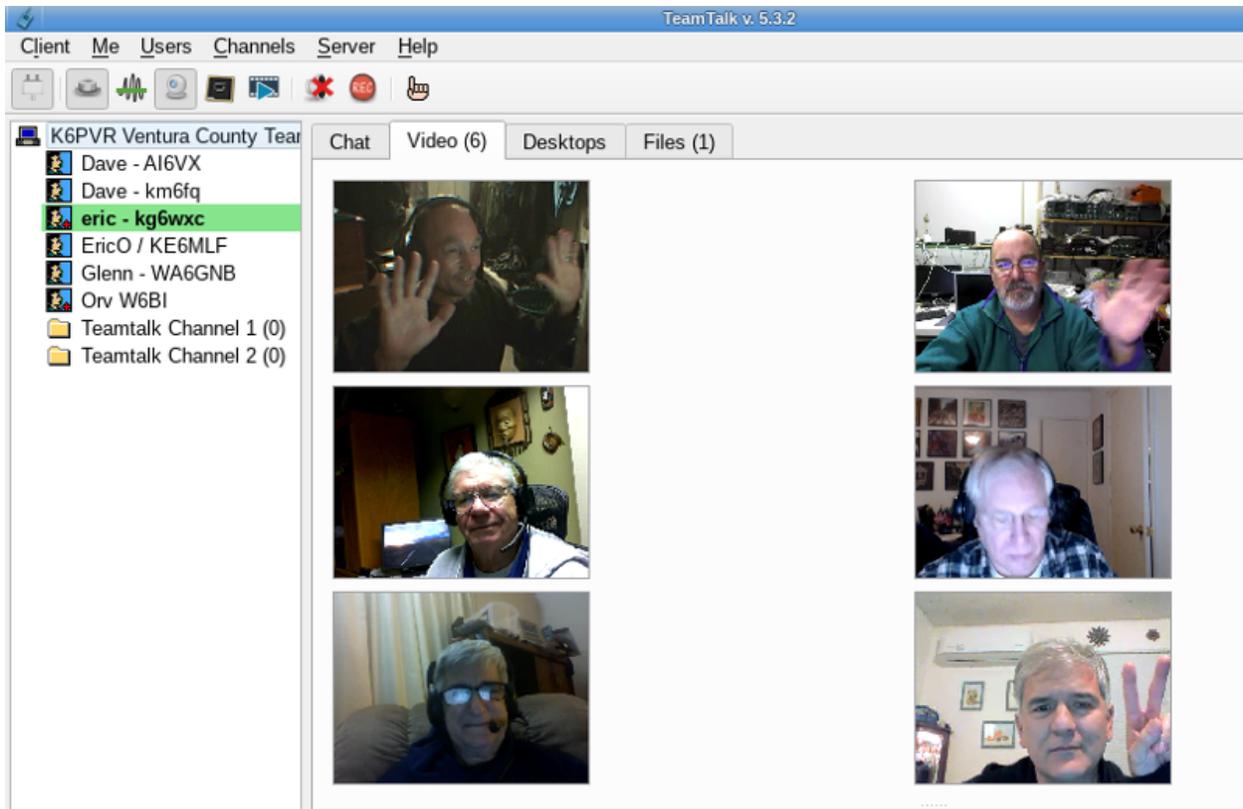
FreeSWITCH provides robust voice and video communication, voicemail, interactive voice response (IVR) menus, user directories, call accounting, screen sharing, chat messaging, call recording, hold music, and many other features that can be implemented as required. It is an extremely flexible communication platform, but you will need specific knowledge and skills in order to install, configure, and manage it as a service.



TeamTalk

TeamTalk is an audio-visual conferencing system which enables people to communicate and share information across the network. It is often classified as *freeware*, but the TeamTalk server is proprietary and its source code is not publicly available. During a conference users talk through their computer microphone, see others via their webcams, create instant messages, share files, and show desktop applications. The TeamTalk software package bundles the client and server programs, so any computer may play the role of client or server.

Voice and video conversations happen in channels or rooms, and a single server can host multiple rooms. While participating in a channel, users can write text messages in the *Chat* tab, view *AV* webcam streams in the *Video* tab, see shared applications in the *Desktops* tab, and download files from the *Files* tab. The server owner can specify a wide range of access permissions for each available room. TeamTalk is currently supported on Windows, Linux, MacOS, and Raspberry Pi computers.



19.4 Example VoIP Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	Features	Network Load	Platform	Effort
Asterisk	extensive	medium	lin/mac/rpi	expert
FreePBX	web management	medium	lin/mac/rpi	medium
Linphone	client softphone	small	win/lin/mac/mobile	easy
Mumble	voice + chat	medium	win/lin/mac	medium
FreeSWITCH	PBX + video	medium-large	win/lin/mac/rpi	expert
TeamTalk	video conferencing	large	win/lin/mac/rpi	easy

Link: [AREDN Webpage](#)

VIDEO STREAMING AND SURVEILLANCE

The previous section described how audio and video traffic can be transmitted across an AREDN® network to facilitate communication. Since these multimedia streams are supported on mesh networks, you can also use them for many other tasks. One example, [video surveillance](#), is often helpful during an emergency or event and AREDN® networks can be used to deliver this type of traffic to Emergency Operations Centers. Keep in mind that multimedia traffic incurs a much greater cost in terms of network performance and computing resources, so be sure your mesh network is designed with the appropriate bandwidth to handle this traffic.

The photo below shows a Mobile Command Center (MCC) deployed to support a large event in San Juan Capistrano, California. An estimated 35,000 people attend this annual gathering, and the local RACES (Radio Amateur Civil Emergency Service) team provides realtime video coverage of the parade route for the sheriff's department and emergency response agencies.



More than a dozen high definition IP cameras were collocated at portable AREDN® node sites across the area, and the individual video streams were consolidated on several large displays in the MCC. Orange County Sheriff's Administrator Sgt. Joseph Cope commented, "This mesh camera system provided by RACES members was a valuable tool for our command staff. The parade was the safest in years. As we were taking the calls, we could see the activity occurring in realtime. Incredibly, there was only one arrest for fighting, which just happened to take place in the camera's view."

20.1 IP Video Cameras



IP video cameras may have a fixed direction and focus, or they may be remote controlled PTZ (Pan,

Tilt, Zoom) models. The cost and features for video cameras vary widely. On the low end is a very inexpensive Raspberry Pi Zero computer having an integrated camera, shown here next to the Ubiquiti Bullet radio. On the high end are the ruggedized commercial PTZ (Pan, Tilt, Zoom) cameras which can cost hundreds of dollars, shown here with the bubble dome and infrared LEDs.

Many IP cameras stream video using [Real Time Streaming Protocol \(RTSP\)](#) in which missing packets are simply skipped during video display. It can be challenging to determine the URL of an RTSP stream, but there is a handy utility at [ispyconnect](#), as well as packet capture utilities such as [Wire-shark](#), which may help. Frequently a camera supports multiple RTSP URLs each with a different resolution, so you can advertise any of them as a service on an AREDN® node as required. Recently more cameras support [ONVIF \(Open Network Video Interface Forum\)](#), which is a set of protocols and standards that includes RTSP. It supports camera discovery and PTZ camera control.

A 1920x1080 resolution video stream at 60 frames/second can consume up to eight megabits/second of network bandwidth. Few AREDN® networks can consistently support that load, but lower frame rates reduce the required bandwidth proportionally. Typically 720p at 10 frames per second is more than adequate for video surveillance.

IP cameras with an Ethernet port are preferred in order to simplify network connectivity and ensure adequate data transfer speeds. Configure the camera to obtain a mesh IP address from the node, and reserve the address for that camera in the node's DHCP settings so you have a consistent way to connect to it. A camera with PoE support is also very useful as this simplifies site cabling.

Some cameras are easier than others to configure and deploy, so be sure to research them carefully before investing in expensive camera hardware.

20.2 Video Display Software

The software described in this section can help you to provision video surveillance services on your network. The following list is not comprehensive or complete but represents a sample of the types of software that may be available for services on your network. Primarily programs with open source licenses were included in this list, although software with proprietary licenses can also be used successfully.

20.2.1 iSpy

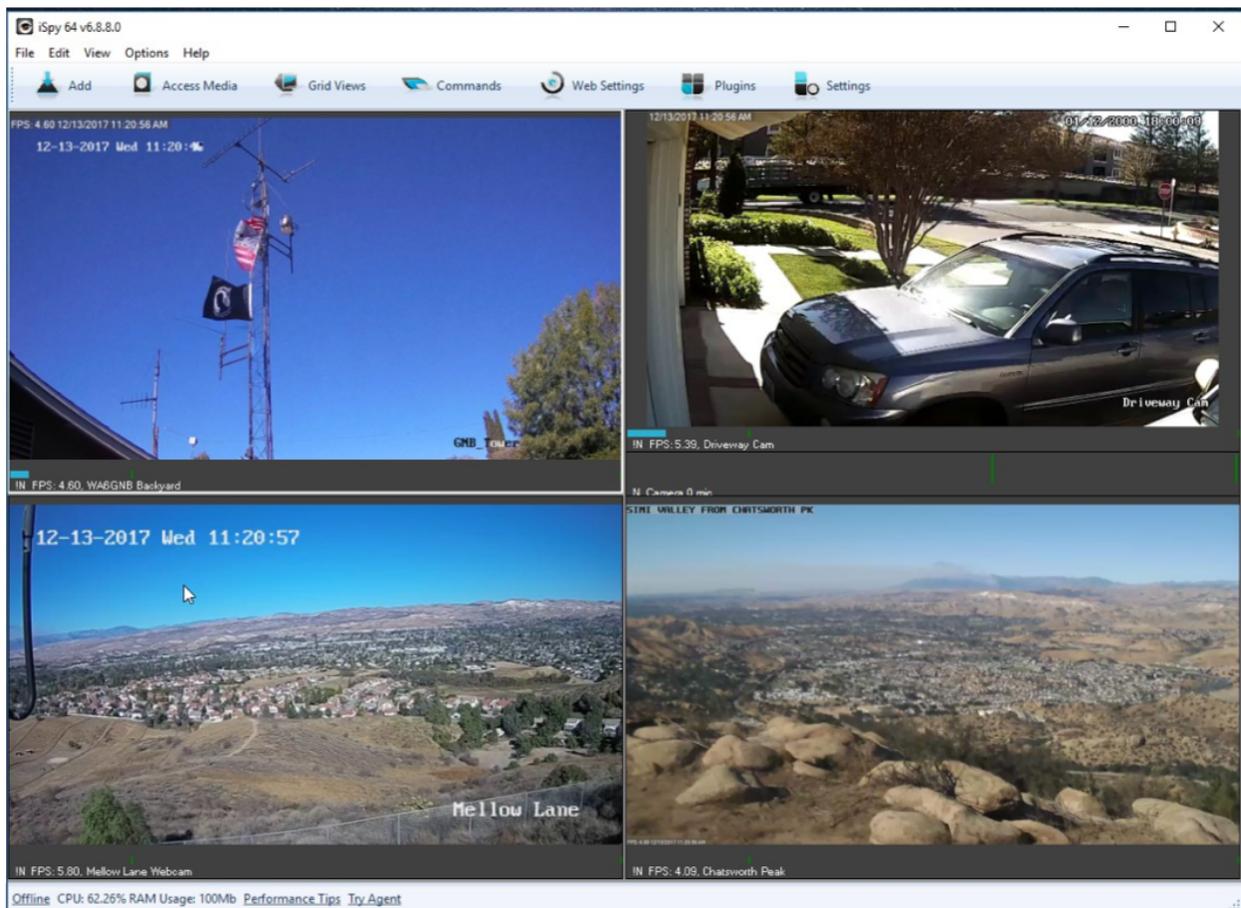
iSpy is a popular video management package for Microsoft Windows computers. It is certified on Windows 7 and above but may work on other systems that support the [.NetV4 Framework](#). iSpy runs as a Windows program with a local user interface (UI) accessible on the computer on which it was installed. Additional services may be available after paying a subscription fee. Parts of the program are licensed under [LGPLv3](#), while other portions are proprietary.

The Windows program provides a “surface” or workspace where you add and configure multiple cameras or microphones. You can then monitor and interact with them to display live video or listen

to live audio from network devices. Multimedia streams can be recorded locally for future use, and PTZ cameras can be manipulated with controls in the UI. Motion detection can also be configured, which provides a method for automatically recording multimedia snippets when specific events occur.

iSpy can connect to IP cameras using MJPEG or JPEG sources. It also supports camera connections using MP4, ASF, or RTSP, which it accomplishes through a VLC plugin after [Videolan](#) software is installed. VLC requires usernames and passwords directly in the URL, so you must enter them in clear text as in this example: `http://admin:password@192.168.1.4/video.asf`.

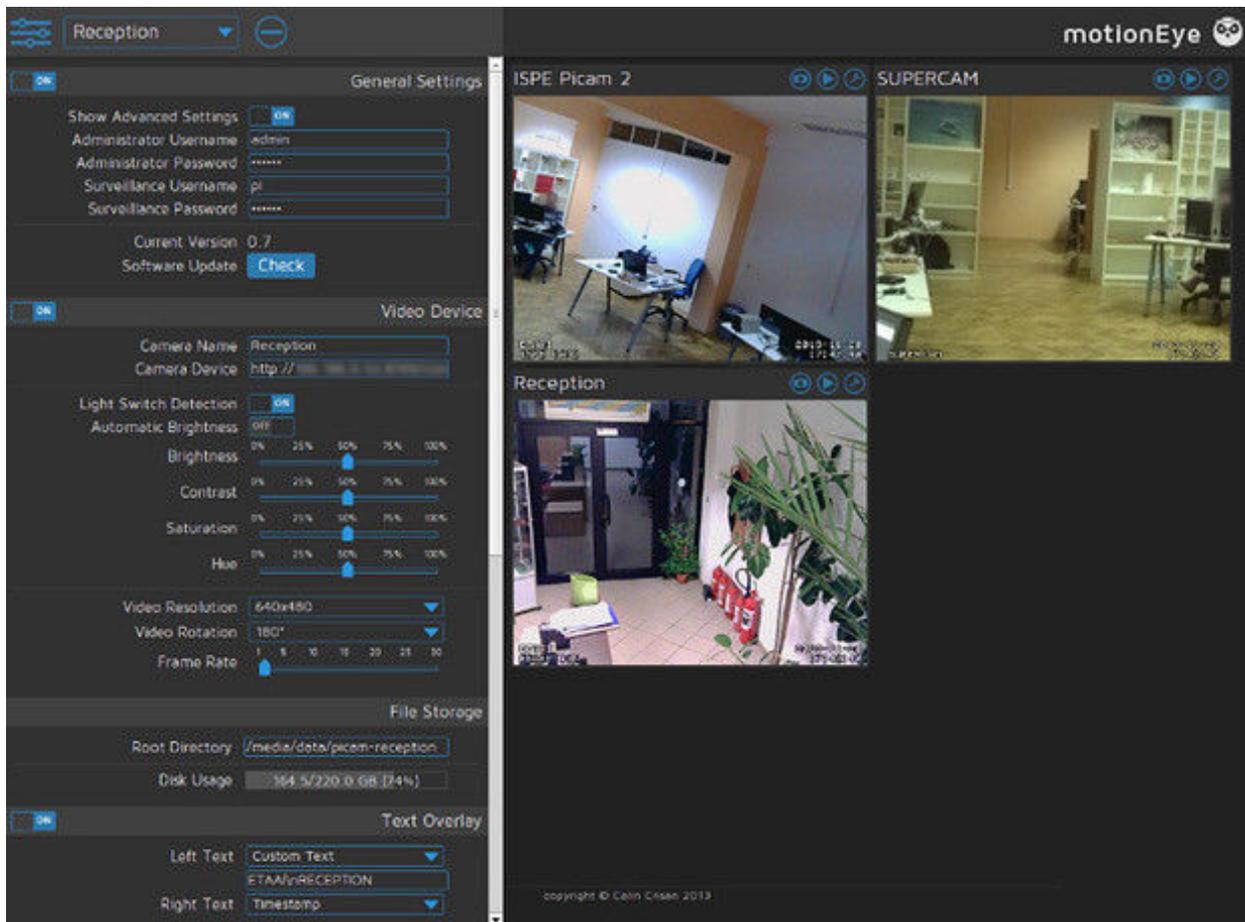
In the lower right video stream on the iSpy display below you can see the smoke plume from the 2017 [Thomas Fire](#) in California, which was recorded by a camera on the local AREDN® network. For additional information about iSpy, visit this link: [iSpy](#).



20.2.2 MotionEye

MotionEye is a lightweight video display program which runs on Linux and Raspberry Pi computers. It can connect to a variety of USB or IP cameras, and it has the ability to display video streams in a grid format accessible by any web browser on the mesh network. Authentication as a regular user or an administrator will display different menu options: view options for regular users or full administrative control for admin users.

The backend [Motion](#) engine is built to provide robust motion detection and event triggering. It also enables custom scripts to extend its features, for example to print the system temperature and update it every ten seconds on the display. Many AREDN® operators implement MotionEye on low-power portable Raspberry Pi computers, and the [MotionEyeOS distro](#) installs the operating system with all dependencies on this platform. For additional information about MotionEye, visit this link: [MotionEye](#)

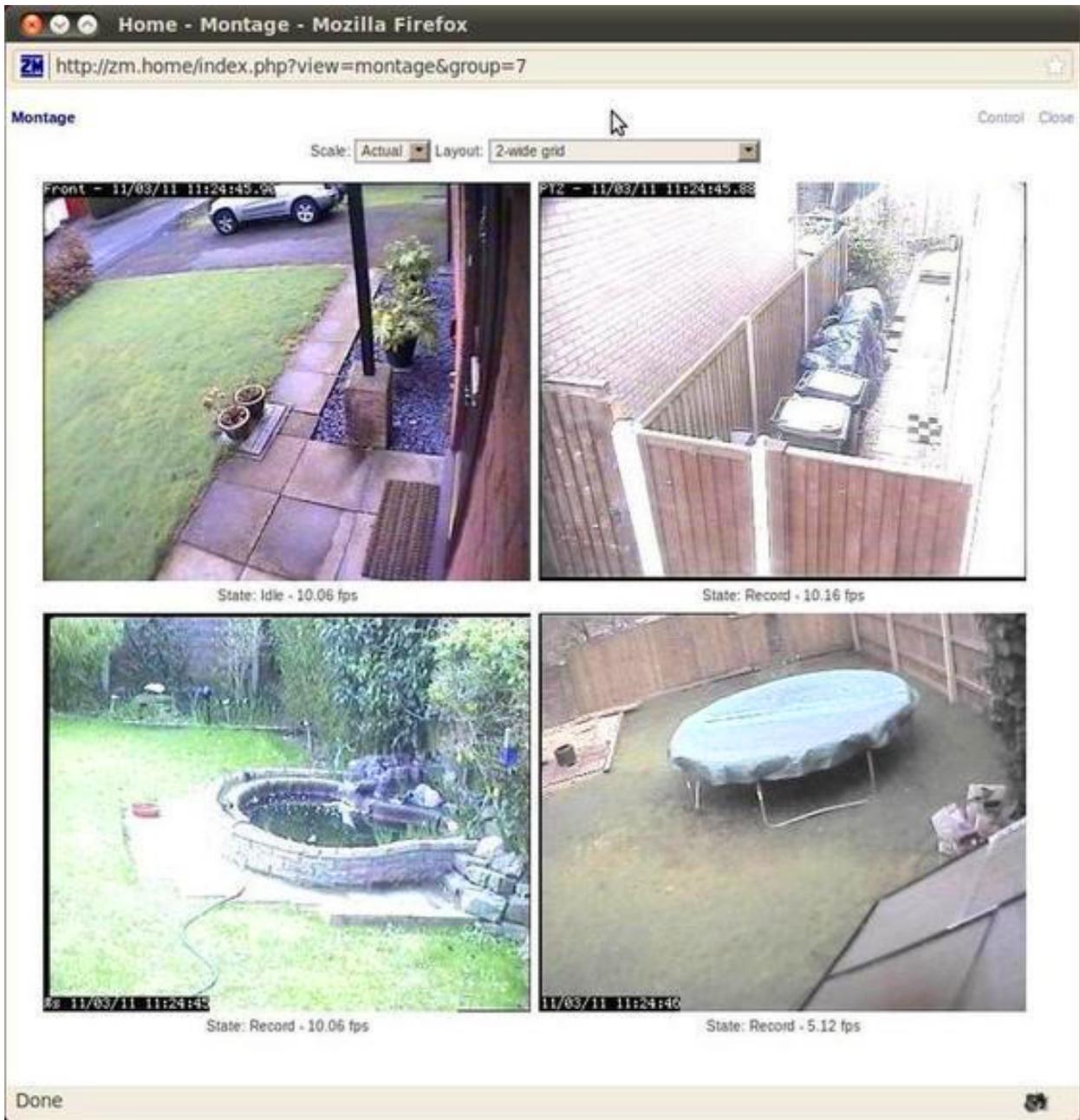


20.2.3 ZoneMinder

ZoneMinder is a full-featured video package which runs on Linux computers. Its display is accessible across the mesh network by web browser. IP cameras are supported which use MJPEG streams or an interface to JPEG images. Camera connections can be configured for monitoring, recording, motion detection, or a combination of these.

The ZoneMinder name comes from the fact that it allows administrators to define “zones” or regions of an image, each with different motion detection sensitivity levels. During motion detection, each frame is compared with previous frames and checked for differences. If the amount of change is greater than a specified percentage, an event will be triggered which can capture recordings, send email alerts, or execute external programs. ZoneMinder has extensive features for filtering and comparing video images, which can be useful for monitoring a high traffic area with a single point of interest such as an entry door next to a busy walkway.

This robust feature set comes at the cost of some administrative complexity, making ZoneMinder a good candidate for operators with skills and experience in Linux and video systems. Its open design and the ability to execute external programs makes ZoneMinder very flexible for integration with other systems. For additional information about ZoneMinder, visit this link: [ZoneMinder](#).

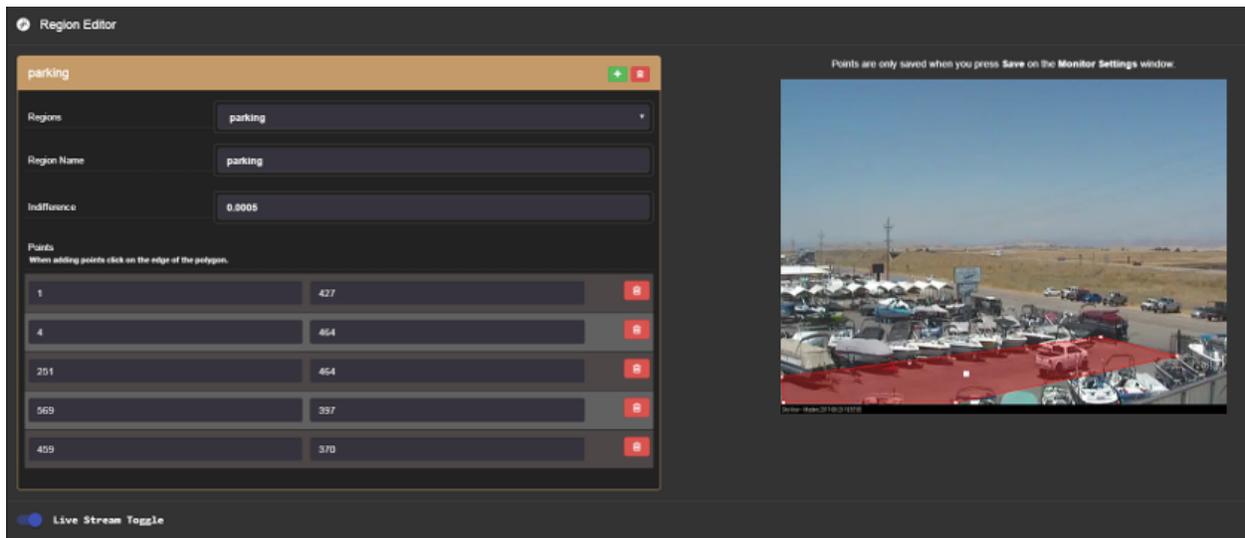


20.2.4 Shinobi

Shinobi is a fairly recent video project which implements current methods of streaming for the web. It supports legacy MJPEG/JPEG, FLV, and RTSP streams as well as the newer HLS and Websocket methods. The web browser interface (UI) is clean and responsive, which renders well on tablets and mobile devices. It is designed for ease of navigation, with dropdown and pop-up menus for snapshots, video recording, event lists, and configuration options.

ONVIF (Open Network Video Interface Forum) compliance allows Shinobi to provide PTZ camera controls. Motion detection is accomplished through plugins, with regions configured in the web UI, so if you do not require motion detection you can conserve resources by not adding it to your system. There are three user levels which provide delegation of authority: Superuser, Admin, and Sub-account. Superusers control system settings and create Admin accounts, which control camera settings and manage Sub-accounts and Groups. Sub-accounts have limited privileges and camera profiles can be shared by Group members.

Shinobi tends to conserve computing resources fairly well, so more cameras or higher resolution streams could be supported on a server. The image below shows how motion detection regions are defined, in this case to monitor traffic along an access road to a parking area. For additional information about Shinobi, visit this link: [Shinobi](#).



20.3 Example Video Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	License	System Load	Platform	Effort
iSpy	freemium	large	windows	easy
MotionEye	open source	medium	lin/rpi	easy
ZoneMinder	open source	large	linux	expert
Shinobi	free for <i>NC</i> use	medium	lin/mac	medium

NC ~ non-commercial

Link: [AREDN Webpage](#)

NETWORK MANAGEMENT TOOLS

There are several service programs that can assist in visualizing or mapping an AREDN® network, as well as for viewing local RF conditions near your node. Some of these programs are discussed below.

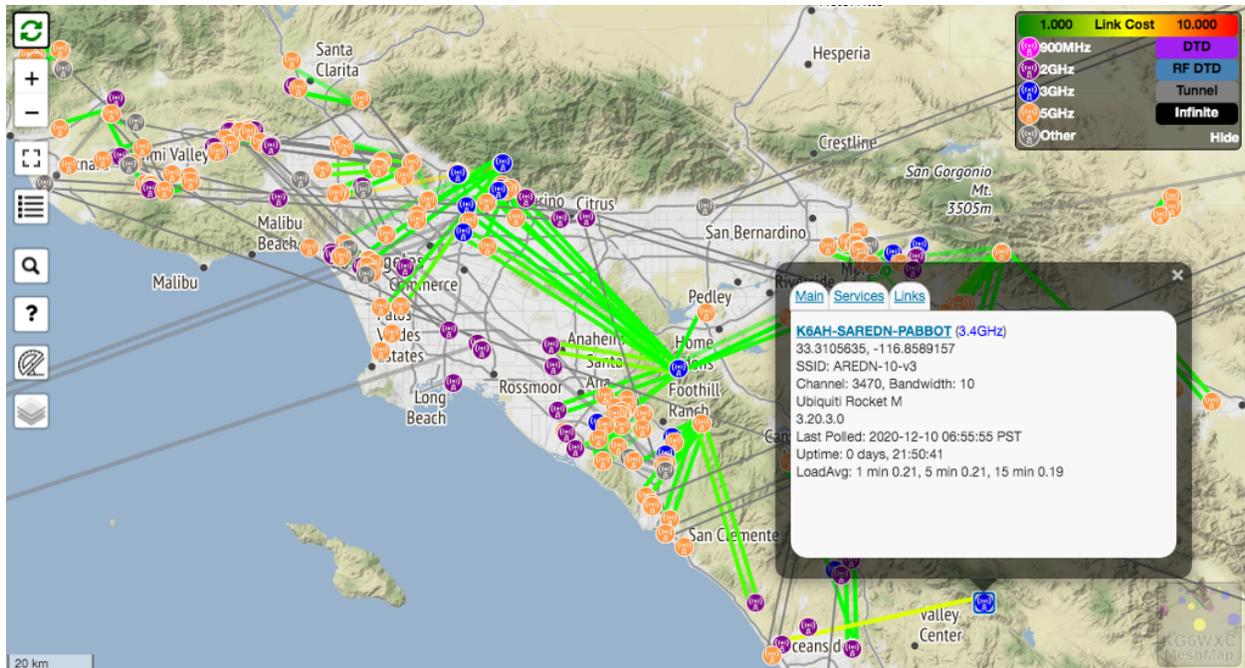
21.1 Manage Extra Static Routes

There may be cases when you need to create extra static routes to control the flow of network traffic through your node. You can maintain your extra routes by entering them into the `/etc/aredn_include/static_routes` file. You must login to your node at the command line and use the `vi` editor to manage the routes in this file. A helpful example is provided in the file, and you can view the [OpenWRT Static Routes](#) page for additional information about managing static routes.

21.2 KG6WXC MeshMap Network Visualizer

[Eric KG6WXC](#) created this useful tool and makes it available as an open source project. MeshMap can be installed on any mesh services computer having [LAMP](#) software, which allows it to run on a Raspberry Pi in your shack or in the field. MeshMap runs continuously and discovers/polls live nodes to display their current configuration, services, and network link information. It maintains a persistent database of all nodes that have been discovered.

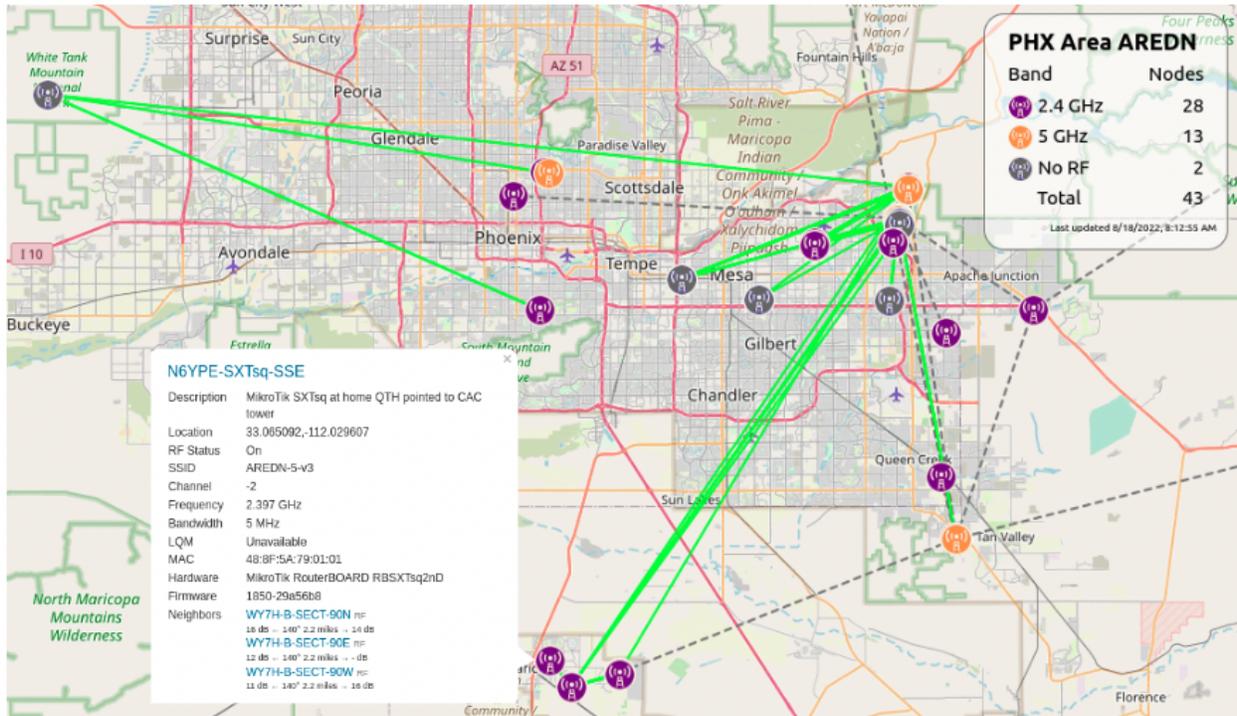
For additional information visit this link: [KG6WXC MeshMap](#).



21.3 KP4MSR MeshMap Network Visualizer

Manuel KP4MSR originally created this software for the Puerto Rico AREDN® network, with a current fork and rewrite of the code maintained by Tim KN6PLV. This program does *not* run continuously and does *not* maintain a persistent database of nodes, so it is less resource-intensive on the network. Once the static pages are built, it can be run on any device with a web server, including on a node with enough free memory.

For additional information visit this link: [KN6PLV MeshMap](#).



21.4 AREDN® Prometheus Exporter

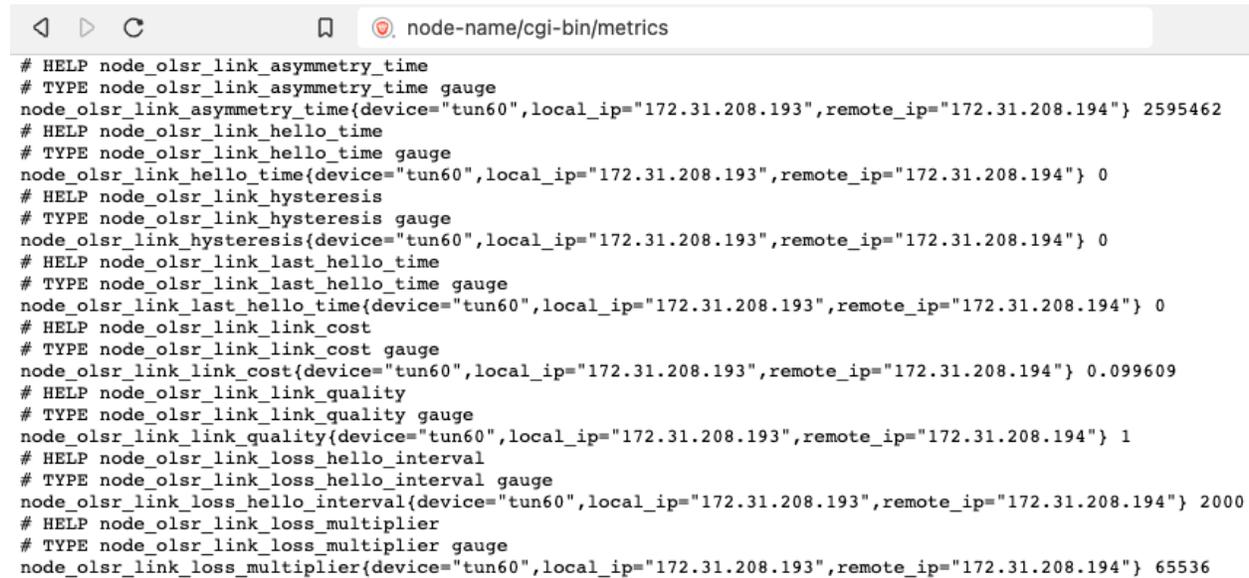
Prometheus is an open-source monitoring and alerting toolkit which collects and stores metrics as time series data. At given intervals it can collect metrics from AREDN® nodes having the prometheus-exporter package installed. Prometheus evaluates rule expressions, displays the results, and can trigger alerts when specified conditions are detected.

AREDN® metrics in the prometheus-exporter package include the following:

- Node details (name, model, firmware, description, Lat/Lon, grid square, band, channel, width, frequency, SSID)
- Memory, storage, CPU, and networking metrics
- RF metrics (signal, noise, MSC rate, TX/RX packets/rates)
- LQM metrics
- OLSR link info

In order for Prometheus to pull metrics from a node it will use the following target URL: `http://<NODE>.local.mesh/cgi-bin/metrics` and metrics are returned by the node as standard

text/plain content. Minimal node resources are required to support Prometheus data collection since the node only uses minimal resources whenever this URL is queried.

A screenshot of a web browser window. The address bar shows 'node-name/cgi-bin/metrics'. The page content displays Prometheus metrics for an AREDN node. The metrics are listed in a plain text format, including help text, metric type, and the metric value. The metrics shown are: node_olsr_link_asymmetry_time (2595462), node_olsr_link_hello_time (0), node_olsr_link_hysteresis (0), node_olsr_link_last_hello_time (0), node_olsr_link_link_cost (0.099609), node_olsr_link_link_quality (1), node_olsr_link_loss_hello_interval (2000), and node_olsr_link_loss_multiplier (65536).

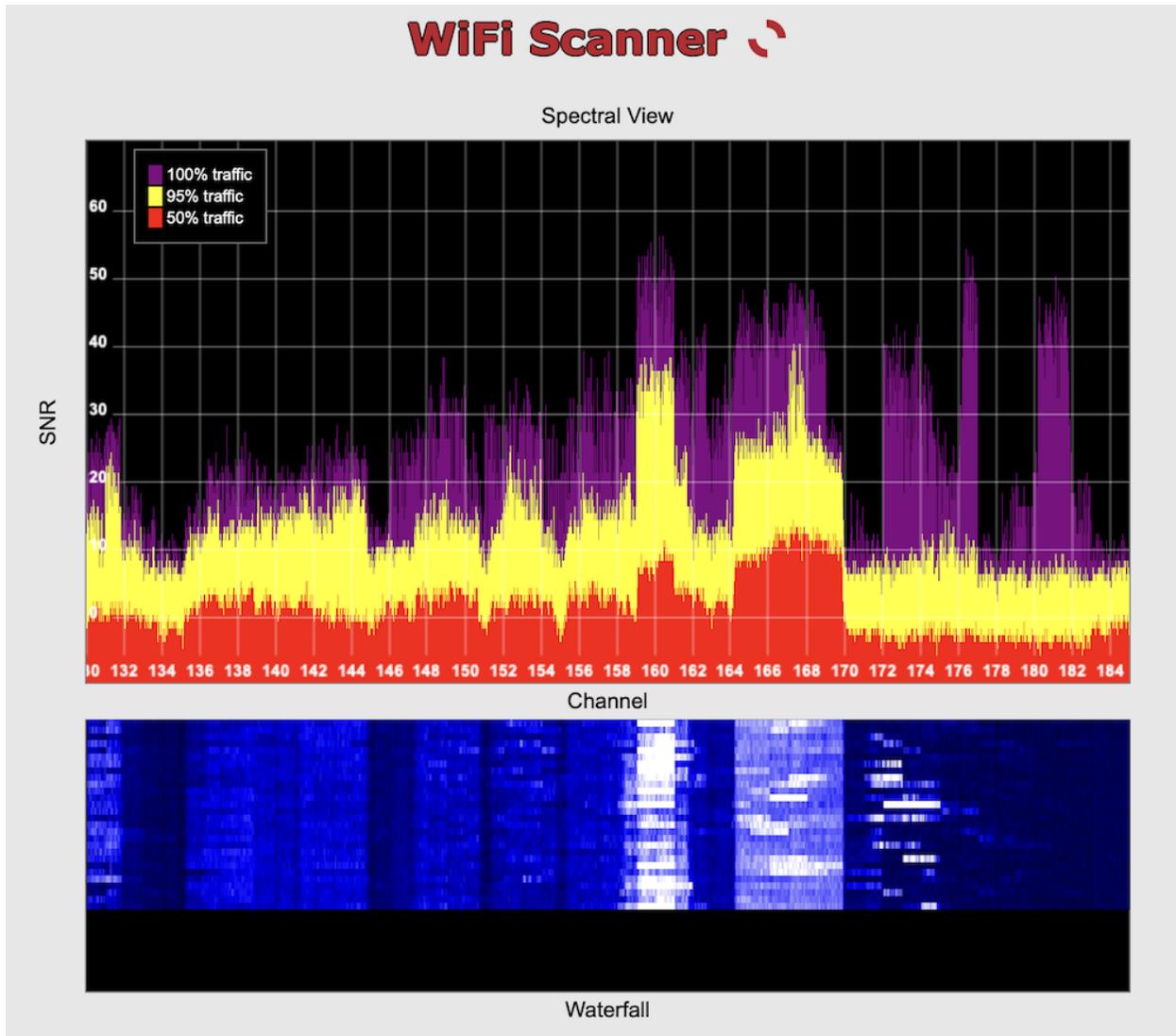
```
# HELP node_olsr_link_asymmetry_time
# TYPE node_olsr_link_asymmetry_time gauge
node_olsr_link_asymmetry_time{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 2595462
# HELP node_olsr_link_hello_time
# TYPE node_olsr_link_hello_time gauge
node_olsr_link_hello_time{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 0
# HELP node_olsr_link_hysteresis
# TYPE node_olsr_link_hysteresis gauge
node_olsr_link_hysteresis{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 0
# HELP node_olsr_link_last_hello_time
# TYPE node_olsr_link_last_hello_time gauge
node_olsr_link_last_hello_time{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 0
# HELP node_olsr_link_link_cost
# TYPE node_olsr_link_link_cost gauge
node_olsr_link_link_cost{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 0.099609
# HELP node_olsr_link_link_quality
# TYPE node_olsr_link_link_quality gauge
node_olsr_link_link_quality{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 1
# HELP node_olsr_link_loss_hello_interval
# TYPE node_olsr_link_loss_hello_interval gauge
node_olsr_link_loss_hello_interval{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 2000
# HELP node_olsr_link_loss_multiplier
# TYPE node_olsr_link_loss_multiplier gauge
node_olsr_link_loss_multiplier{device="tun60",local_ip="172.31.208.193",remote_ip="172.31.208.194"} 65536
```

The AREDN® `prometheus-exporter` simply makes these metrics available for Prometheus to pull. For additional information about Prometheus itself, visit [their website here](#). The following image shows Prometheus metrics for an AREDN® node being displayed by the [Grafana](#) visualization application.



21.5 KN6PLV Network Waterfall Scanner

Tim KN6PLV created this program to assist with discovering the RF conditions around your node. It is installed as a node package which is available here: [KN6PLV Waterfall](#). Once installed, a new **Waterfall** button will appear on your *Node Status* page if the hardware supports it (the Waterfall scanner is not currently available on AC devices). It will disconnect your node from the mesh while it continuously scans for nearby RF signals, so it does require authentication with the node's login credentials in order to run. The *Spectral View* shows the strength of nearby signals, while the *Waterfall* maintains a record over time of the RF environment. This package does *not* work on node hardware having 802.11ac chipsets.



[Link: AREDN Webpage](#)

COMPUTER AIDED DISPATCH

Computer Aided Dispatch provides an automated way for emergency services agencies to keep track of incidents, activities, information, tasks, messages, and the status of deployed resources. Command staff are able to see the big picture, while at the same time maintaining detailed records of plans and actions for future reference. Deployed resources are able to clearly communicate in realtime, while having much better situational awareness of surrounding events.

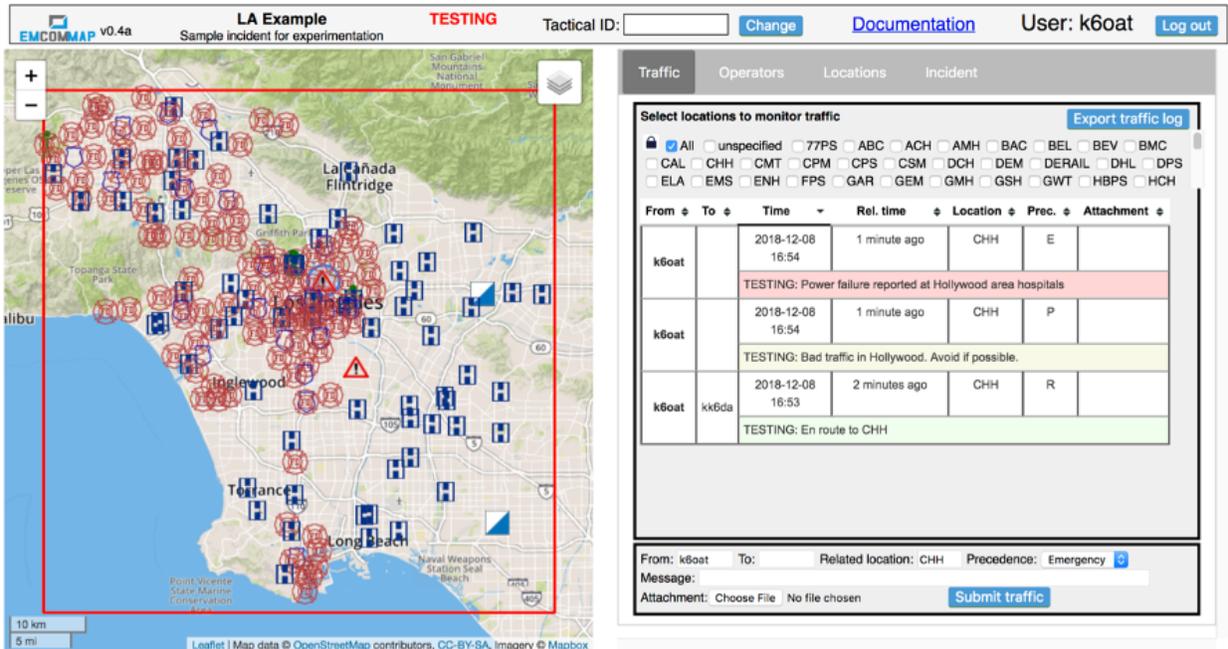
Served agencies have been using Computer Aided Dispatch (CAD) software for quite some time, and it has become their preferred method for managing events and incidents within their jurisdiction. In emergencies when electrical power or mission-critical facilities become unavailable and agencies are forced to operate off-grid, AREDN® operators with portable power for mesh networks and computing resources can bridge the gap by providing CAD (Computer Aided Dispatch) solutions for personnel at key sites.

There is a wide variety of CAD software in use today. Many of the sophisticated commercial packages have integrated automatic vehicle location (AVL) and geographic information systems (GIS) which require large amounts of network bandwidth and dedicated computing resources that might not be accessible during an emergency.

The programs described in this section can help you to provision CAD services for emergency use on your mesh network. The following list is not comprehensive or complete but represents a sample of the types of software that may be available for services on your network. Programs with open source licenses were included in this list, although software with proprietary licenses can also be deployed.

22.1 EmComMap

EmComMap was designed by an Amateur Radio Emergency Service operator for use on AREDN® mesh networks during deployments. It leverages modern technologies for interactive maps and sync-able web browser databases to enable map-based situational awareness and emergency communication across IP networks. Based on this architecture, EmComMap is one of the more mesh-friendly CAD programs with additional features in progress for data distribution.



A specific geographic region is defined within which an incident is in progress, and the location of resources are shown on the map using icons (*Police, Fire Department, Hospital, Government Facility, Incident Command Post, EmComMap Node*). Each map can be zoomed and panned as required to view location details for all deployed resources. Incident information can be defined and updated on the *Incident* tab, while locations are defined and updated on the *Locations* tab. Message traffic is available to all operators across the network on the *Traffic* tab, and operators update their location and status on the *Operators* tab. Open Street Map tiles can be downloaded to the server for standalone operation.

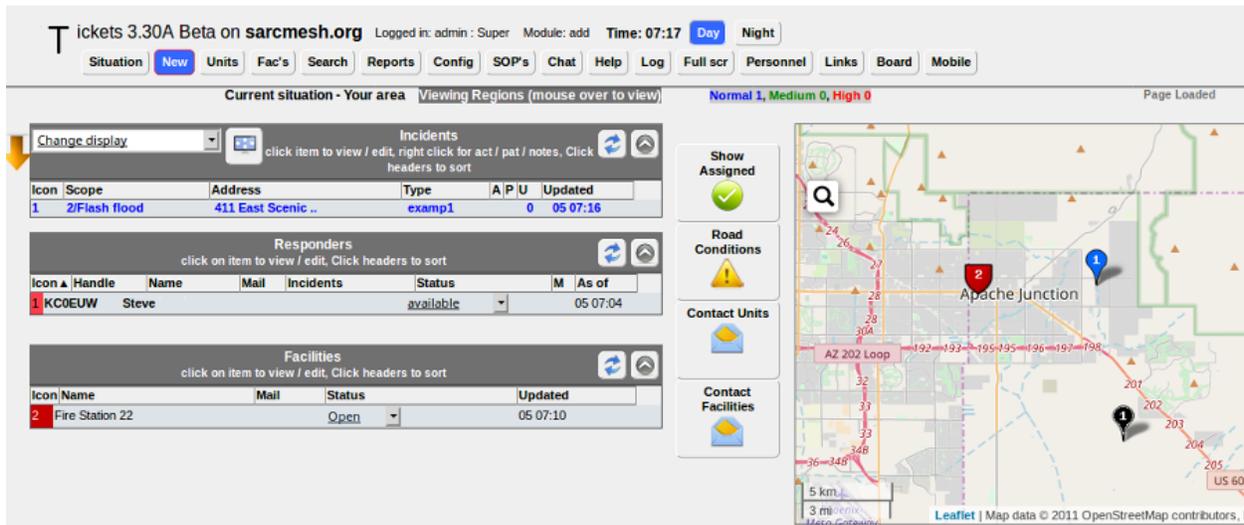
All communications are tracked and can be exported in spreadsheet format for offline use. Message traffic can be filtered to view specific messages for selected locations, and the traffic table can also be sorted for viewing the details based on information in any column. Message severity levels and tactical call signs are supported, and operators are allowed to send messages and report status information on behalf of other users if necessary. EmComMap is a recent program under active development, with continual feature improvements in progress. For additional information about EmComMap, visit this link: [EmComMap](#).

22.2 Open ISES Tickets

The *Open Information Systems for Emergency Services (ISES)* project is a community of software developers, paramedics, EMTs, law enforcement, and fire fighters working to create software and training materials for the emergency service community. They currently offer the *Tickets CAD* system, which has an extensive suite of features that are accessible by web browser from a mesh network server. Any computing platform is capable of running a *Tickets* server if it supports the traditional [LAMP](#), [XAMPP](#), or [MAMP](#) packages.

Tickets presents a situation dashboard showing incidents, responders, and facilities along with a GIS map of their locations. Open Street Map tiles can be downloaded for standalone operation. Clicking any of the controls allows operators to drill into item details, and *Tickets* provides database tracking for a large array of information about each item. The dashboard can be fully integrated with several different functions, including email, chat, routing, and tracking (for example, with [Automatic Packet Reporting System \[APRS\]](#)).

A variety of built-in reports are available which can be viewed, printed, and downloaded for distribution. Standard ICS forms are available for online completion and emailing, and custom *Standard Operating Procedure (SOP)* documents can be integrated for viewing through dashboard links in the web browser. For additional information about *Tickets*, visit this link: [Open ISES Tickets](#).



22.3 Example Computer Aided Dispatch Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	License	System Load	Platform	Effort
EmComMap	open source	small	linux	medium
ISES Tickets	open source	small	win/lin/mac/rpi	medium

Link: [AREDN Webpage](#)

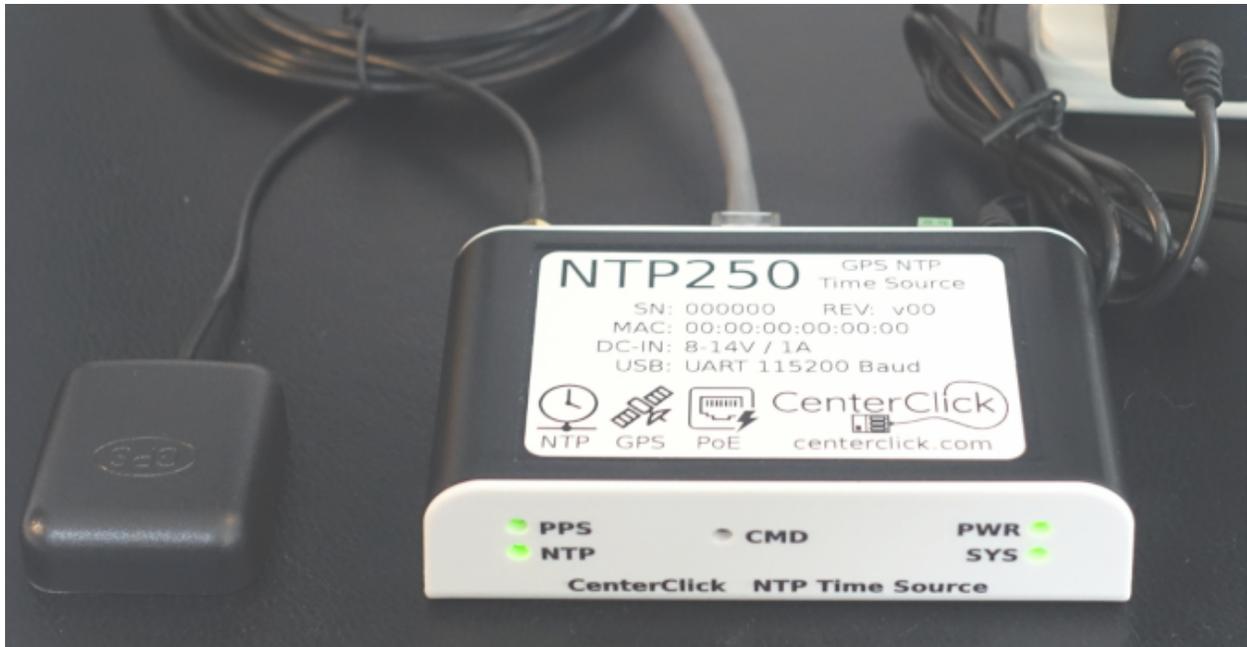
OTHER SERVICES

As mentioned in the *Services Overview*, almost any program that can operate across a peer-to-peer TCP/IP network is a candidate for AREDN® networking. Many useful services have been discussed previously, and this section will list some of the other types of services that you might consider deploying on your mesh network.

23.1 Network Time Services

Although the AREDN® nodes themselves do not depend on network time synchronization, there may be other programs or services running on your mesh network which would benefit from having accurate network time updates. [Network Time Protocol \(NTP\)](#) is a reliable way for networked devices to update their system clocks. This may be especially helpful for devices that do not have an onboard realtime clock, such as Raspberry Pi computers. It may also be important to have accurate timestamps across the network for programs such as MeshChat, email message logging, file timestamps, video surveillance images, and many others.

Most NTP implementations depend on an Internet connection in order to synchronize with upstream time servers. However, it would be more useful to be able to synchronize system clocks in an off-grid situation when AREDN® nodes are deployed during an emergency. One way to accomplish this would be to configure one or more battery powered computers as NTP servers which retrieve upstream time from GPS satellites (*stratum 0*).



Position your portable NTP server so that it maintains a clear view of the sky and can get a fix on as many GPS satellites as possible. In order for NTP to operate properly, each client device must have a reliable connection to the NTP servers on the network. Be sure to locate your NTP servers on reliable high-speed segments of your mesh.



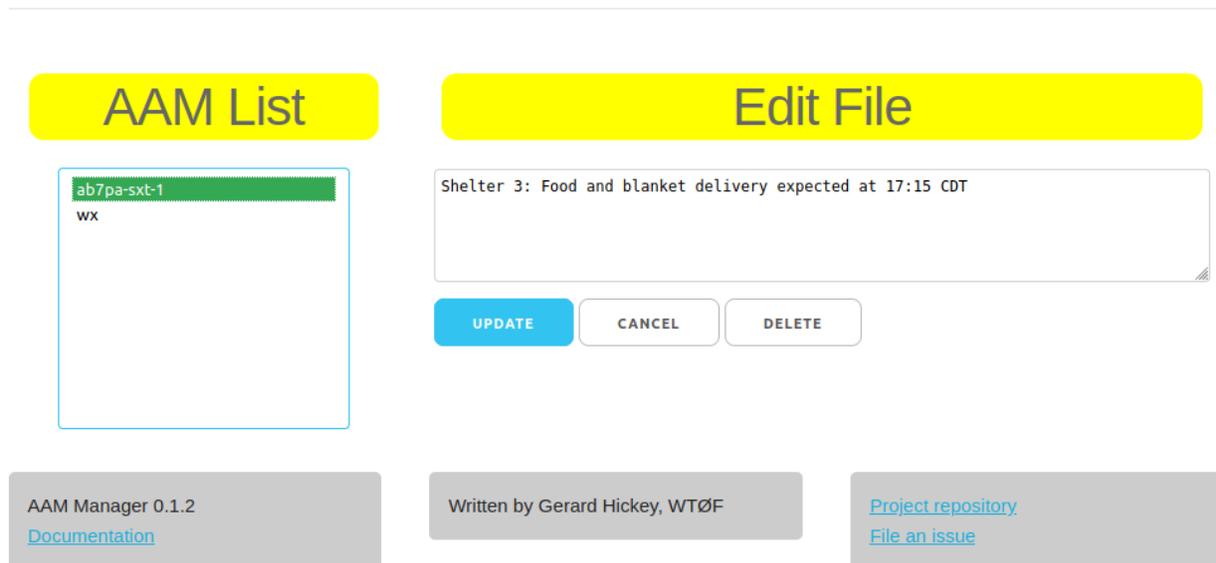
You may choose to purchase an inexpensive off-the-shelf NTP appliance such as those offered by

Centerclick and others. There are also many sources of information for building your own off-grid NTP server (for example, this one using a Raspberry Pi: [G4WNC NTP](#))

23.2 AREDN® Alert Message Manager

AREDN® Alert Messages were explained in the **Getting Started Guide** under the *Node Status* and *Advanced Configuration* sections. The example given there showed the Alert Message source running on a separate LAN-connected web server. It is also possible to provide Alert Messages using an application created by Gerard Hickey (WT0F) which runs directly on a node having adequate storage. The AREDN® Alert Message Manager (*aamm*) uses the node’s web server to provide a web interface for creating, updating, or deleting Alert Messages – as well as actually hosting the message repository on the node itself, so that no external LAN-connected web server is required.

AAM Manager



The two advantages of using this application are 1) having the message management front-end shown above, and 2) having a node-hosted message source which eliminates the need for a separate LAN-connected web server. Alert Messages are presented as shown in the example below on the *Node Status* display.

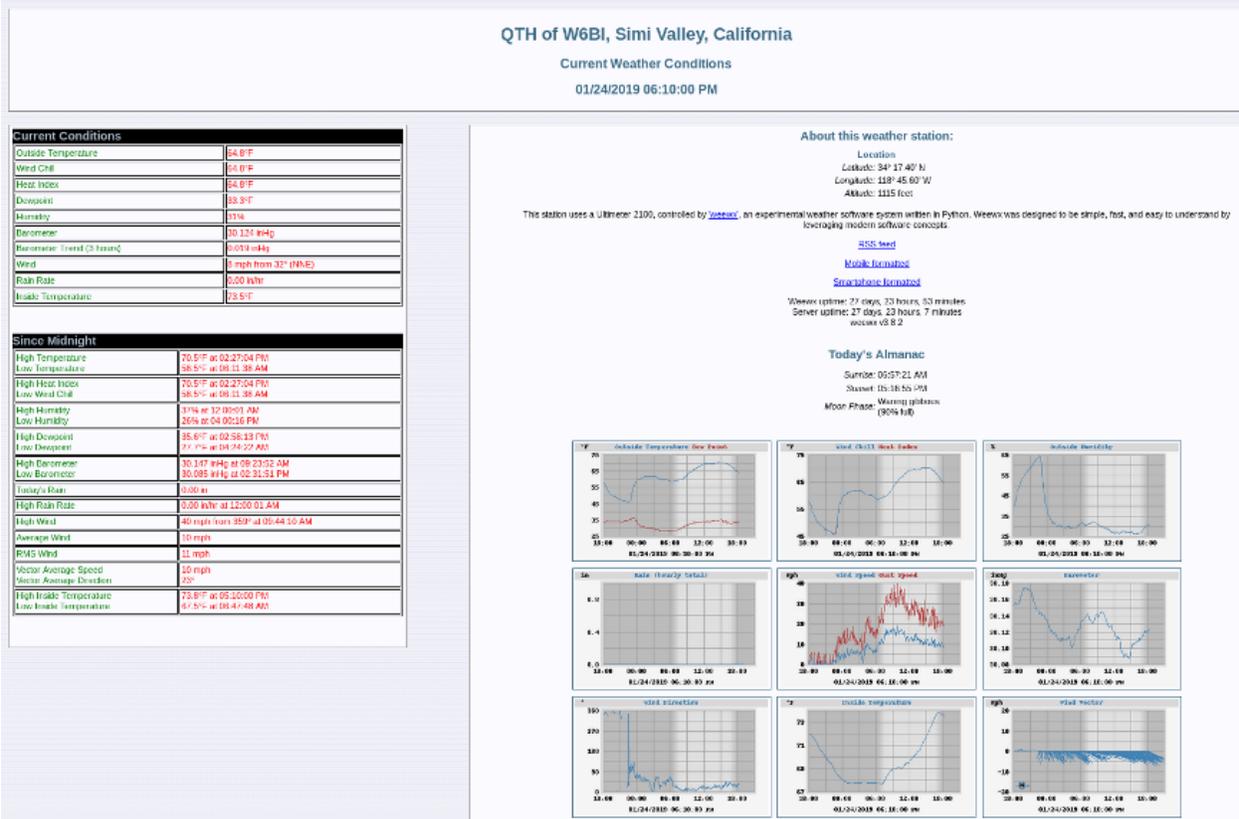
Local Messages:

- **ab7pa-sxt-1:** Shelter 3: Food and blanket delivery expected at 17:15 CDT
- **wx:** A SEVERE THUNDERSTORM WARNING REMAINS IN EFFECT UNTIL 500 PM CDT FOR NORTHERN CASS...SOUTHEASTERN JACKSON AND WEST CENTRAL JOHNSON COUNTIES.

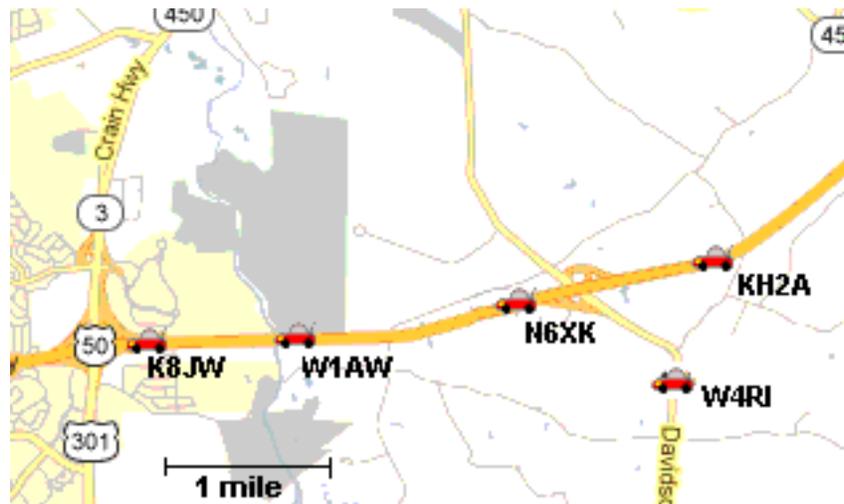
The recipient nodes are configured the same way as described in the **Getting Started Guide** under the *Advanced Configuration* section for AREDN® Alert Messages. For additional information about the AREDN® Alert Message Manager, visit this link: [aamm](#). You may also download and install the latest *aamm* package files [here](#).

23.3 weeWx Weather Service

Many operators have weather stations, as do quite a few repeater sites. If those weather stations can be put on the mesh network, they can provide a valuable overview of weather conditions across a wide area, for example, showing wind speeds and rainfall totals for each location. The *weeWx* package is available for many different operating systems and weather station models. It supports serial, USB, and Ethernet connections to weather stations. For additional information about *weeWx*, visit this link: [weeWx](#).



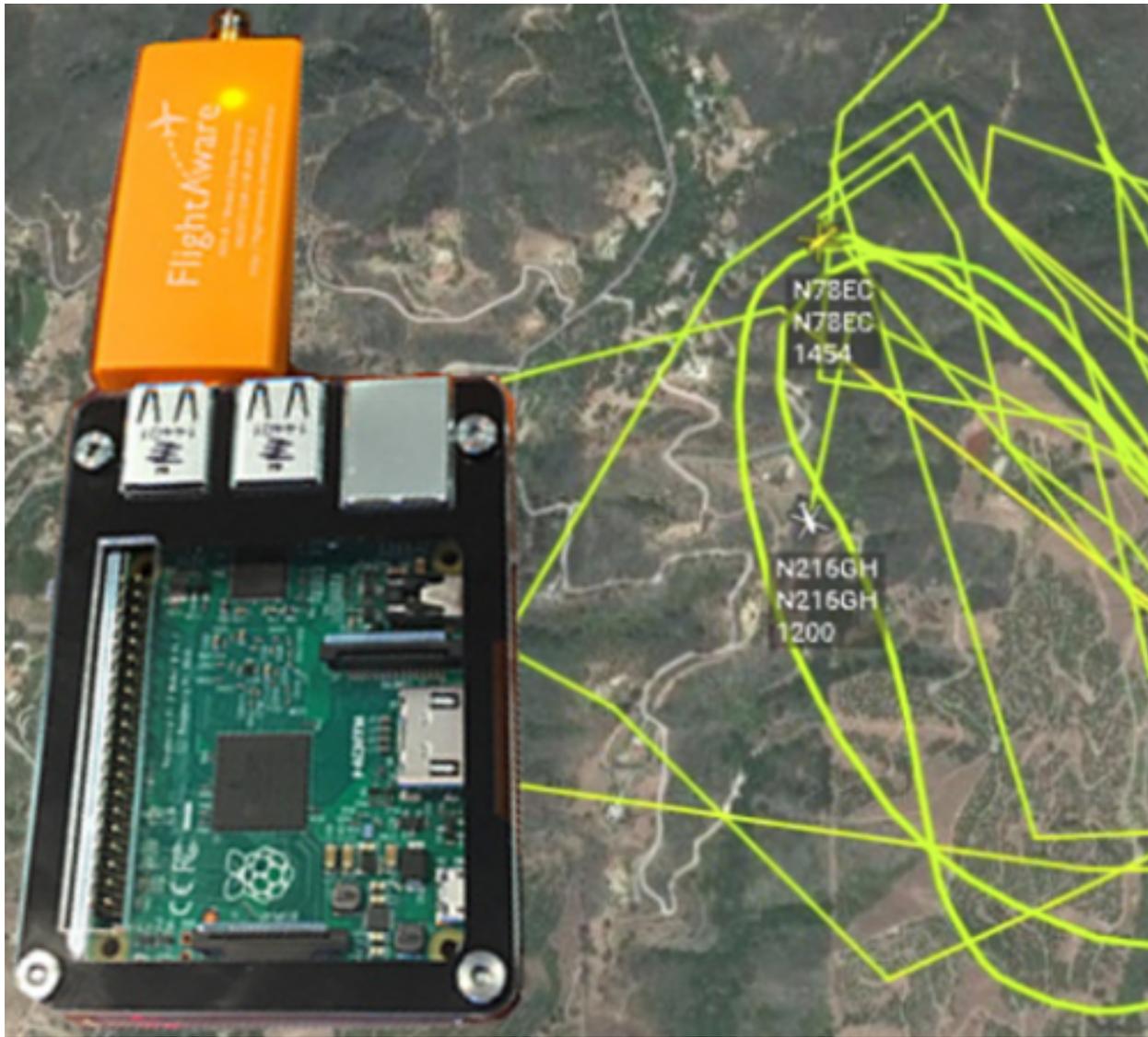
23.4 GPS Tracking Services



Tracking deployed resources is an important task during any emergency. There are many options for monitoring and displaying the GPS locations of tracked resources, two of which are mentioned here.

Many amateur radios and portable locating beacons transmit [Automatic Packet Reporting System \(APRS\)](#) information. It is possible to implement an APRS receiver using inexpensive, battery-powered, portable computers and USB [Software Defined Radios \(SDR\)](#). The details are widely available for building these receivers using Raspberry Pi computers with [Direwolf](#) and [Xastir](#) or [YAAC](#) software.

There may be situations when it would also be helpful to track the locations of aircraft during an emergency. [Automatic Dependent Surveillance-Broadcast \(ADS-B\)](#) information is available which can be captured using portable computers with ADS-B receivers. The following image shows the track of two water tankers dropping fire retardant above Santa Barbara, California, during the 2017 [Thomas Fire](#). This information was displayed across an AREDN® network using an [ADS-B Ground station](#) which was running as a mesh network service.



Depending on the requirements of your specific situation, almost any program that can operate across a peer-to-peer TCP/IP network could be deployed as a service on your mesh network.

Link: [AREDN Webpage](#)

TIPS FOR UPLOADING FIRMWARE

Uploading firmware to an AREDN® node is usually a straightforward process. Follow the procedures documented in the **Downloading AREDN Firmware** section to ensure you have the correct firmware version from the AREDN® website to install on your node. If you experience issues uploading firmware, the following tips may be helpful.

Error message when uploading firmware

If you see an error message displayed when uploading new firmware to your node, verify that you are loading the correct file by referring to the [AREDN Firmware Selector \(AFS\)](#), then you can safely ignore the warning. The file naming standard recently changed from a non-standard naming convention to the standard naming convention used by OpenWRT.

Web browser cache and sessions

One common issue can occur when installing firmware using a web browser. Your computer's browser cache stores data for the URLs that have been visited, but IP addresses and other parameters may change during the install process. It is possible for the cache to contain information that doesn't match the latest settings for the URL, so the browser may block the connection setup and display an `ERR_CONNECTION_RESET` message. Clearing your computer's web browser cache will allow the latest URL settings to be registered so you can continue with the install process.

Instead of a *Connection Reset* message, sometimes a *Bad Gateway* message may appear. This is an [HTTP Status Code](#) that can mean any of several things. Often it indicates a network communication issue between a web browser and a web server. During AREDN® firmware installs you can usually resolve a *Bad Gateway* issue by doing one or more of the following things:

- Refresh or Reload the URL for your node.
- Clear your browser cache and delete cookies.
- Close your browser and restart a new session.
- Use a different web browser program or a *Safe Mode / Incognito* browser window.
- Unplug and reconnect the Ethernet cable from your computer to ensure that your machine has received a new DHCP IP address on the same subnet as the node's updated IP.

PXE Server

If you are using a PXE server to provide your device with an IP address and a new firmware image, be sure to allow the PXE server through your computer's firewall. If the PXE server does not display any activity when you begin your firmware install, check your firewall settings. On the Windows control panel, for example, click *Advanced Settings* and look through the "Inbound Rules" to see if a rule exists for the PXE server. If a rule exists, make sure to "allow connection" for both private and public networks. If no rule exists, create a new rule allowing connection for both public and private networks.

24.1 Tips for Upgrading Firmware

Upgrading an AREDN® node is accomplished using the *Setup > Administration > Firmware Update* feature on the node's web interface. Follow the procedures documented in the **Downloading AREDN Firmware** section to ensure you have the correct firmware version from the AREDN® website to install on your node.

Note: Currently there are two Mikrotik devices which require that the standard firmware compatibility checks be disabled in order to upgrade from version 3.22.12.0 or older to a newer firmware version. The specific devices are shown in the **Supported Devices** list on the AREDN® website with a *Status* of *danger-upgrade*, and the notes at the bottom of that page explain what is required. You must first install the **Dangerous Upgrade package** (the **ipk** file) which will disable the firmware compatibility checks. After this package is installed on your node you can perform a normal firmware upgrade (for example) from 3.22.12.0 to 3.23.4.0.

In rare cases the upgrade process can fail due to lack of node resources, but such a failure will leave the node running its previous firmware version. The following tips help ensure that memory utilization is at a minimum on the node.

Try to Load Local Firmware

The **Load Local Firmware** option is described in the *Configuration Deep Dive > Administration* documentation. This involves using a file copy utility on your computer to copy the firmware file to a specific directory and filename on your node. Once the new firmware file is available on the node, you can click the *Load Local Firmware* button to start the install process.

Tips for legacy nodes with low memory (32mb)

Legacy equipment with only 32mb of memory may require more effort to upgrade. Be sure not to use these types of devices at sites which are difficult to access.

- Before starting the firmware upgrade on low memory devices, it may be necessary to stop, disable, or uninstall extra packages such as MeshChat, SNMP, and IperfSpeed. The goal of this step is to keep those processes from using RAM memory and to free as much RAM as possible before the upgrade. Rebooting the node before beginning the

upgrade will ensure that RAM utilization is at a minimum.

- You may also want to stop node programs or services that are not needed during the upgrade. For example, you can telnet or ssh to the node and type the command `wifi down` to free the memory used by this driver.
- You may need to try the `sysupgrade` procedure several times before it succeeds. Be patient and keep trying.
- Get everything ready to do the upgrade, then do a fresh reboot of the node and immediately start the `sysupgrade` process before the node has time to initialize services which use memory.
- Use command line access to copy the `sysupgrade.bin` image to the `/tmp` directory on the node, then run the `sysupgrade` process manually from the command line on the node. Note that AREDN® nodes use port 2222 for secure copy and secure shell access.

Execute the following commands from a Linux computer:

```
my-computer:$ scp -P 2222 aredn-firmware-filename.bin root@192.
→168.1.1:/tmp
my-computer:$ ssh -p 2222 root@192.168.1.1
~~~~~ after logging into the node with ssh ~~~~~
node:# sysupgrade /tmp/aredn-firmware-filename.bin
```

To transfer the image from a Windows computer you can use a *Secure Copy* program such as [WinSCP](#). Then use a terminal program such as [PuTTY](#) to connect to the node via ssh or telnet in order to run the `sysupgrade` command shown as the last line above.

- As a last resort, use the First Install procedure to load the `factory.bin` firmware image to the node. This procedure is described in the *First Install* sections of **Installing AREDN Firmware**.

24.2 Tips for Downgrading Firmware

Downgrading AREDN® firmware is typically accomplished using the same procedure as for uploading firmware to your node. You are simply uploading a previous version of the firmware rather than the latest version.

However, there is a difference if you are downgrading the firmware on a node which previously used a different target architecture. As explained in the **Downloading AREDN Firmware** section, the legacy `ar71xx` target has been retired and replaced by the `ath79` target. For example, you may have a node that was previously running an `ar71xx` firmware version but you installed the latest Stable Release or Nightly Build which upgraded it to an `ath79` firmware target. In this case you will need to do a fresh First Install using the legacy architecture's firmware.

1. Use the [AREDN Firmware Selector](#) to download the previous release's install files. For example, if your Ubiquiti Rocket M5 XW is currently running version 3.23.4.0, then download the files required for a First Install from release 3.22.12.0 which used *ar71xx* (as shown below).

Download AREDN Firmware for your Device

Type the name or model of your device, then select a stable build (ie. 3.22.12.0) or the nightly "snapshot" build (ie. 2050-781425a).

Ubiquiti Rocket M XW 3.22.12.0 ▾

About this build

Model: **Ubiquiti Rocket M XW**
Platform: ar71xx/generic
Version: 3.22.12.0 (r11427-9ce6aa9d8d)
Date: 2022-12-14 15:51:08
OpenWrt 
Info: 

Download an image

 **FACTORY**

Use a Factory image to flash a router for the first time. Depending on the device, this could be TFTP, PXE, or other special instructions. See docs.arednmesh.org for details.
sha256sum: 8281c6229d45f6b904c65dba2fbd311f3639c182d2eb5d8480375a44e8d9452b

 **SYSUPGRADE**

Use a Sysupgrade image to update a router that already runs AREDN. The image can be used with the AREDN web interface.
sha256sum: b16891737f1ecacbcfd74d6dc5c3c14723a1f069d928255a3b6a3e5e19afd9cc

2. Review the **Installing AREDN Firmware** documentation and follow the steps for the *First Install* procedure that is appropriate for your node model.
 - For Ubiquiti and TP-LINK models you will be uploading the *FACTORY* firmware.
 - For Mikrotik models you will boot using the *KERNEL* file (which you rename to *rb.elf*) and then immediately apply the *SYSUPGRADE* firmware image.
 - For GL.iNet models you will use the [recovery procedure](#) to upload the *SYSUPGRADE* firmware image.

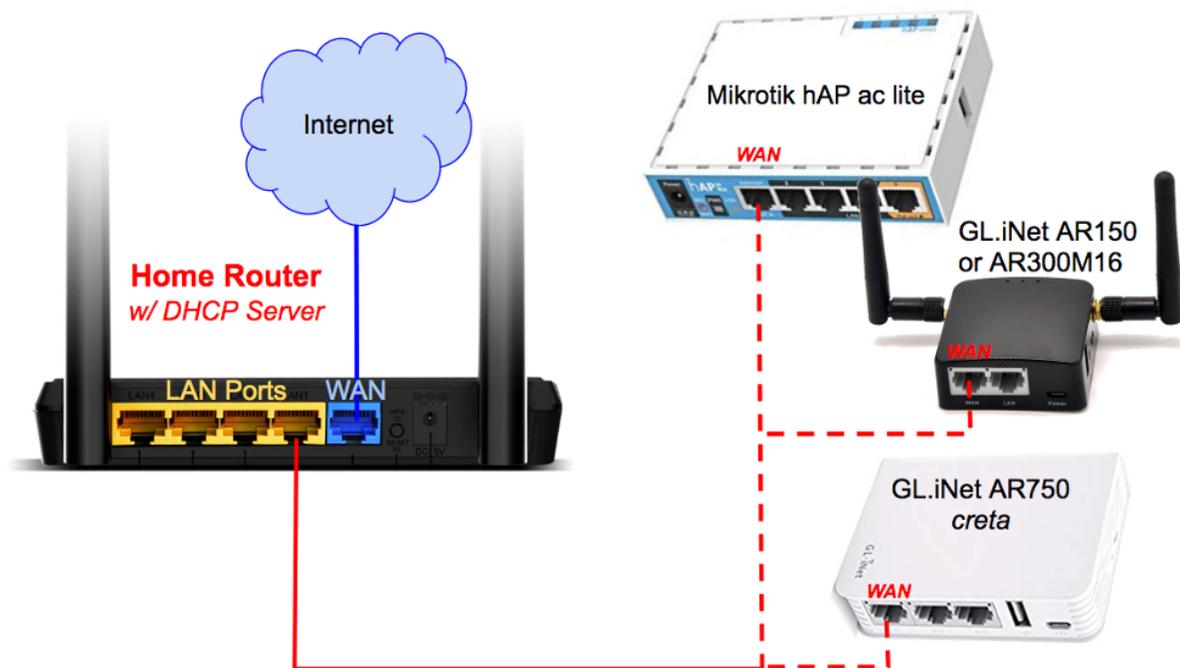
Another possible way to downgrade firmware between architectures is to enable **Dangerous Upgrade** under the *Advanced Configuration* settings. Setting this to *ON* will disable the normal firmware compatibility checks that are done automatically during the firmware install process. This should allow your node to install a firmware image that uses a legacy architecture.

After downgrading your node's firmware you will then continue the process for entering your call-sign and configuring the node's settings, as explained in the **Basic Setup** section.

[Link: AREDN Webpage](#)

CONNECTING NODES TO HOME ROUTERS

There are several indoor AREDN® nodes that have more than one Ethernet port, including the *Mikrotik hAP ac lite* as well as the *GL.iNet AR150*, *AR300M16*, and *AR750 Creta*. The AREDN® firmware running on these types of nodes has the WAN port preconfigured for connecting to the Internet. You can get the latest information about the specific port configured as the node's WAN port from the AREDN® website here: [Ethernet Port Usage](#)



When you connect the node's WAN port to one of the LAN ports on your home router, the node's WAN should receive an IP address on your home network from the router's **DHCP** server. Alternatively you can reserve an IP address in your home network range and assign the static IP to the node's WAN through the **Basic Settings** page on your node. There are many sources of information about basic **home networking** which will not be duplicated here, but feel free to familiarize yourself with IP networking through reading and research.

Once you have connected your node to your home router, Internet access will be available to the node

itself as well as to any of the devices connected to the node's LAN network. It is not recommended to allow Internet access through your node from other Mesh connected nodes, therefore be sure to leave *“Allow others to use my WAN”* unchecked. If you do not want any of your node's LAN connected devices to access the Internet either, you can check *“Prevent LAN devices from accessing WAN”*.

Link: [AREDN Webpage](#)

POWER OVER ETHERNET (POE)

The phrase **Power over Ethernet (PoE)** encompasses any of several different standards and methods for passing DC power over twisted-pair Ethernet cabling. The advantage of PoE is that it allows a single cable to carry both data and power to your devices, and several AREDN® supported devices can be powered using PoE.

This section of the documentation provides a high-level overview for those who are not already familiar with this concept. You do not need to be an expert in *Power over Ethernet* technology, but it may help to be aware of a few concepts in case you run into these terms when researching PoE switches or injectors.

26.1 Passive PoE

At the present time, all of the PoE radios supported by AREDN® require the use of **Passive PoE**. In a *Passive PoE* system the power source does not negotiate voltage or wattage requirements with the powered device. *Passive PoE* power sources simply supply a specific voltage constantly, up to the maximum current limit that the power source allows.

The primary message of this section is to encourage you to read the manufacturer's data sheet carefully for the hardware that you will be deploying. Pay particular attention to the specifications for **Input Voltage** and **Maximum Power Consumption**. The allowed voltage ranges and maximum power consumption for AREDN® radios will vary by hardware model as shown in the comparison below.

Example Data Sheet Info

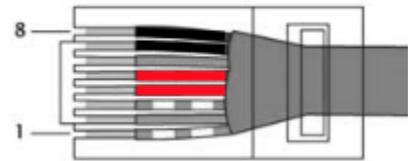
	Mikrotik LHG-5nD	Ubiquiti PowerBeam-M5-400
Input Voltage	11 - 30 vdc Passive PoE	11 - 28 vdc Passive PoE
Power Consumption	6W	6W

You simply need to determine what voltage range your equipment accepts and then use a power source that constantly supplies a voltage within that range. For example, the Mikrotik device in

the table above can accept a range between 11 and 30 volts, while Ubiquiti devices typically accept between 11 and 28 volts. This means that you could use either a 12v or 24v source to power these devices because both are within the acceptable voltage range for these radios.

In this example, both models have a *Maximum Power Consumption* of 6W. This means that you can expect a maximum current draw of approximately 500mA if you use a 12v battery to power them. If you use a 24v source then you would expect a maximum current draw of only 250mA.

Passive PoE commonly uses the same Ethernet cable wires as are used in the IEEE 802.3af **Mode B standard**, but that is the only similarity between *Passive PoE* and that standard. *Passive PoE* uses separate wire pairs to carry data or power as shown below. DC positive is carried on pins 4-5, DC negative is carried on pins 7-8, while data is carried on pins 1-2 and 3-6.



Cable Pins	Wire Pair Usage	Data	Power
Pin 1	RX+		
Pin 2	RX-		
Pin 3	TX+		
Pin 4			DC+
Pin 5			DC+
Pin 6	TX-		
Pin 7			DC-
Pin 8			DC-

You should not need to concern yourself with the various IEEE 802.3 standards that may be used for other types of PoE equipment. The radios currently supported by AREDN® do not use standards such as *802.3af*, *802.3at*, *802.3bt*, *PoE+*, *4PPoE*, or *Ultra PoE*. There is a wealth of information on the Internet if you decide to learn more about these other standards.

Be aware that it should not damage your AREDN® device if you connect it to an 802.3af/at switch or PSE (power sourcing equipment). The only consequence would be that the device will not be powered, since switches using the other standards will not send power if they do not detect a compatible device.

[Link: AREDN Webpage](#)

LINK QUALITY MANAGER (LQM)

Contributor: Tim Wilkerson KN6PLV

AREDN® mesh networks often lack the bandwidth you might expect. Here we look at what may be happening, a proposal to fix it, and results from these fixes.

27.1 Introduction

Low SNR links between nodes break the Linux “auto distance” algorithm resulting in poor bandwidth utilization on node links. This document describes *Link Quality Manager* which can be enabled on nodes to better manage RF links. Typically 3x bandwidth improvements have been observed, but much higher improvements are possible.

27.2 Expected link speed vs. actual link speeds

We’ve all pointed an AREDN® node at another node, found the sweet spot for the best SNR, and then been underwhelmed by how much bandwidth there seems to be. WiFi is famous for over-reporting how much bandwidth is available vs. what you actually get (think 1/6th in many cases), but somehow we all expect a 2 mile link with SNR > 20 to do more than 1 Mbps. Unfortunately actual performance data is difficult to come by, and experiments to see what might improve are difficult to coordinate. Several theories are described below.

27.3 Performance theories

27.3.1 Hidden nodes

The classic problem with CSMA networks like AREDN® is that of “hidden nodes”. CSMA works by nodes listening for transmitters before themselves transmitting. This can fail when nodes are

spread out so only some nodes can hear others resulting in many transmitting simultaneously, corrupting data, requiring retransmissions, and wasting bandwidth.

While this could be a real issue for AREDN® networks, it only has an effect when there are a lot of collisions. For there to be a lot of collisions there has to be a lot of traffic ... but AREDN® networks are largely idle. Probably the biggest traffic generator is OLSRD, and on the SF Bay Area network (for example) OLSR accounts for only a few kilobytes/second. Statistically it is difficult to ascribe the bandwidth problems to hidden nodes.

27.3.2 Bandwidth decimation

Bandwidth decimation occurs when too many radios are using the same channel at similar locations, and so rather than adding bandwidth each radio ends up with a share of the original.

As noted above, AREDN® doesn't currently use much of its available bandwidth. There is some overhead in having multiple radios on the same channel, even if they are not very active, but it is not significant. Compare this to a home wifi setup: if many devices are constantly using a lot of bandwidth then it is definitely noticeable.

Checking the status of various omnidirectional antennas on the SF Bay Area network (for example), there are rarely more than five neighbors on each node and never more than ten. If everyone were transmitting constantly there would be a problem, but this does not occur often.

27.3.3 CSMA vs. TDMA

TDMA is more efficient than CSMA and avoids many of its problems. However, Linux currently has no implementation of the protocol; it is restricted to proprietary radios. This means that AREDN® as currently envisaged cannot support TDMA. While TDMA radios can have a place in an AREDN® network, they seem better suited for backbone operation.

That said, racing to embrace TDMA without understanding why the current CSMA network is failing is problematic. If hidden nodes aren't the issue, and network utilization is too low for bandwidth decimation, how would TDMA fix this? What actually is the problem?

27.4 Alternate theory

27.4.1 Coverage Class

The WiFi standard (IEEE Std 802.11TM-2007, Part 11) briefly discusses "Coverage Classes". This defines the air time propagation for the wifi signal. For in-home networks this parameter is unimportant as devices are close together, and the actual time it takes for packets to transfer between devices is essentially zero. However, for long distance networks this parameter becomes more important. A wifi signal propagates at approximately one mile every five microseconds. This seems

fast but it quickly becomes significant especially over 10 and 20 mile links. Coverage Class is designed to account for wifi propagation time. If the Coverage Class is too high, devices will wait longer than necessary to retransmit failed packets. If the class is too low, devices will retransmit packets unnecessarily. Having an appropriate Coverage Class for a long distance network link is very important for optimal performance.

27.4.2 Auto-distance

AREDN® provides a “Distance to FARTHEST Neighbor” setting which, indirectly, allows the Coverage Class to be set (the Linux kernel calculates the coverage class from the distance setting). An “auto” option is provided and enabled by default. “Auto” uses a “dynamic ack” algorithm in the kernel which automatically adjusts the Coverage Class. The adjustment is based on the timing of packets sent to and acknowledged from other devices. The class will always be large enough to handle the most distant device.

AREDN® is an open, ad hoc, network allowing any node to associate with any other node as long as it uses the same channel and bandwidth. This results in distant nodes with very low SNRs being associated with each other. Unfortunately the dynamic-ack algorithm does not know that these links are essentially unusable, but it still adjusts the Coverage Class to accommodate them. The result is a higher Coverage Class than is required for optimal network operation, resulting in longer delays in packet retransmission. This compounds the already increased retransmissions inherent in longer links and further reduces the throughput.

27.5 Link Quality Manager (LQM)

The *Link Quality Manager* can be enabled on any node running 3.22.12.0 or newer firmware. It runs in the background to evaluate RF links and automatically take the following actions:

1. Blocks radio links which are also DtD links
2. Blocks radio links which have too low an SNR
3. Blocks radio links which are too distant
4. Blocks radio links with too many retransmission errors
5. Sets the node’s Coverage Class based on the most distant non-blocked node that is a direct, routable neighbor.

27.5.1 What does this mean?

1. Occasionally nodes are directly connected (DtD) to collocated nodes which are also using the same channel. Although the DtD link should be preferred by OLSRD, LQM ensures that any radio link between DtD nodes is always ignored.
2. LQM ignores links with SNR too low to be useful. The application uses adjustable settings to accomplish this: drop below the minimum SNR and the link is blocked until the SNR is above the activate level. The hysteresis avoids links bouncing in and out of a blocked state. This stops OLSR from using poor links.
3. LQM limits how far a node can be from a neighbor and still have a reliable link, even if there is a high SNR. The more distant a node, the lower the throughput of the link. In addition, the total throughput on a node is affected by the most distant node it communicates with. LQM automatically determines the distance between nodes using the latitude and longitude information available from each node's sysinfo.json api.
4. Some links can have high SNR, not be far away, but still have terrible performance due to excessive retransmission errors. While some retransmissions are to be expected, if this rate becomes large then performance suffers. LQM blocks links with poor link quality.
5. LQM disables automatic distance detection and takes over the job of managing the Coverage Class. LQM evaluates the non-blocked links and determines whether there is at least one route which uses this link. It then selects the link with the largest distance and uses this to calculate the Coverage Class.

The *Link Quality Manager* refreshes its state every minute and adjusts the blocked nodes and Coverage Class calculations. The *Neighbor Status* display shows the state of each link, while the LQM settings can be adjusted on the **Basic Setup** or **Advanced Configuration** displays.

27.5.2 What LQM does not do

LQM blocks nodes by blocking traffic from the appropriate MAC addresses. What it does not do is prevent nodes from associating with the radio. It would be ideal to either ban “poorly performing” nodes from associating with a radio, or alternatively telling the node not to associate with distant radios. However, the ad-hoc wifi mode used in AREDN® does not currently support this.

27.6 Test Results

LQM has been deployed and tested on a number of links with various radio environments and properties, both in the San Francisco Bay Area as well as in Southern California. Early feedback from these experiments have helped to refine and improve LQM and the results presented below are from version 0.4.

In the tables below we list various links of different lengths which were tested with and without LQM. Where possible the signal-to-noise ratio at both ends of the link were noted. Bandwidths were measured using multiple runs of *iperf3* in both directions (the results separated by slashes). Additional notes highlight information relevant to the nodes and related tests.

27.6.1 SF Bay Area Network

Link (miles)	Distance	SNR	No LQM (Mbps)	LQM With (Mbps)	LQM Notes
2		25/28	0.282/2.79	13.3/20.6	Channel 177, very congested in this area
2		36/31	38.8/32/6	50.4/50.9	Channel 173, 20 MHz, no congestion

27.6.2 Southern California Network

Link distance (miles)	Dis-	SNR	No LQM (Mbps)	LQM With (Mbps)	Notes
4			6.4/6.3	11.3/11.0	Links running from single node to 3 other nodes with similar distances, some congestion
5			11.4/11.1	16.0/15.8	
5			9.2/9.1	16.7/16.4	
11			2.5/2.2	9.6/9.4	
20			4.9/4.7	4.8/4.6	
34			0.7/0.6	0.7/0.7	Congested site with a mix of short and very long links

These results yield the following conclusions. LQM never negatively affects bandwidth, but the positive effect can be very large. The only result where there was no measurable improvement was at a site having a mixture of many long and short distance links. As expected, the very long 34 mile link negatively impacted all other links on that radio. Improvements of 47x was observed in one case (which was verified multiple times) and it occurred in a crowded, noisy environment. More typical improvements were around 3x.

27.7 Conclusions

Experiments with the *Link Quality Manager* have demonstrated that we can improve the throughput on links by a significant amount without making physical changes to the network. Improvements of 3x bandwidth are common and in many cases much more is observed.

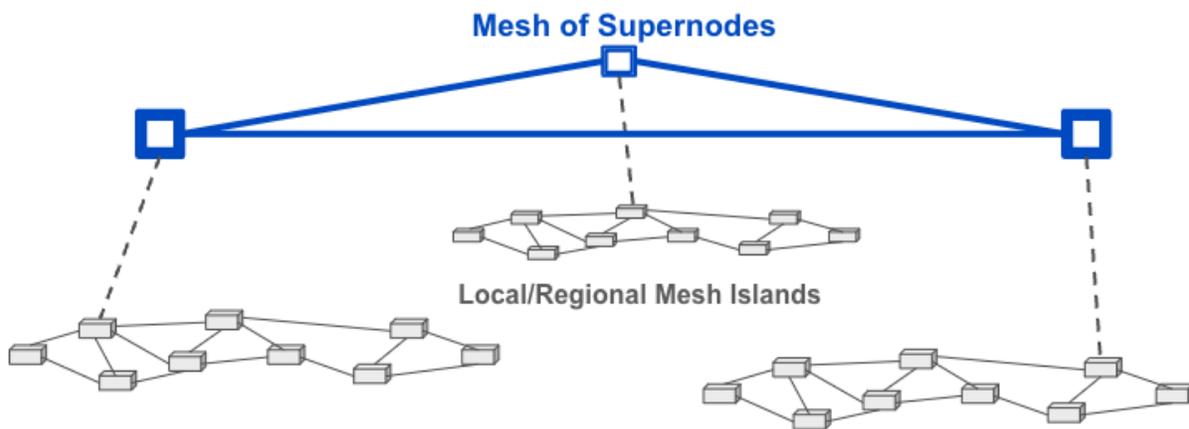
LQM also blocks paths in the network which are marginal, either due to excessive distance, poor SNR, or high retransmissions. We expect that by blocking poorly performing links the entire network will be more stable and performant.

Nodes with a mix of long and short links showed less improvement because the radio is optimized for the longer link distance. This increases retransmissions delays on the shorter links, reducing the throughput and lowering overall node performance. It might be better to use two radios at those sites to offload the longer links.

[Link: AREDN Webpage](#)

CONFIGURING A SUPERNODE

Supernodes are a way to link multiple mesh island networks in a safe and efficient way. A Supernode network is a high-level mesh network — **super** meaning “*above or higher.*” The Supernode network sits above the isolated mesh networks and provides connectivity without increasing the routing load on the local networks. Supernodes do not merge networks into one big mesh but instead isolate connections between discrete meshes. For further information see the *Supernode Architecture* section of the **Network Topologies** topic in the **Network Design Guide**.



28.1 Criteria for Deploying a Supernode

Before you consider deploying a Supernode, make sure you can adequately support the level of uptime that is desired for the Cloud Mesh network.

1. Fast unlimited Internet connection. Fiber is preferable. Low latency between Supernodes is important as is available bandwidth. A Supernode can easily transfer 1 teraByte of data every month, so an unmetered connection is best.

2. Uptime stability. The Supernode should be up 99.999% of the time. The location should ideally have backup power and network connectivity, both to the Internet and to the local mesh.
3. Solid local mesh connectivity. As this is the path for all traffic between your local mesh and every other mesh, the connection to your mesh should be at a high-bandwidth location. If you are deploying a Supernode with any sort of high-bandwidth backbone, the Supernode should be connected to the backbone.

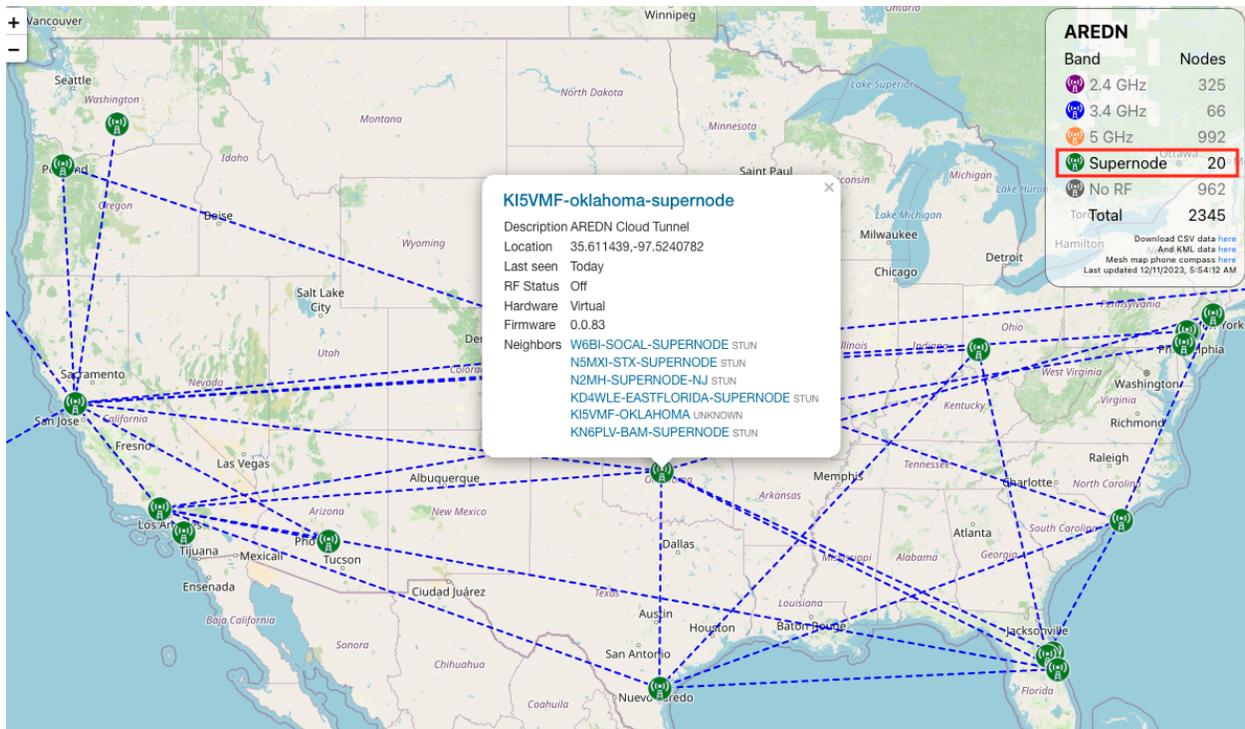
28.2 Coordinating Supernode Deployments

Because Supernodes use the [OLSR \(Optimized Link State Routing\)](#) protocol, multiple Supernodes can be connected to each other, each operating as a peer of the others. A local network can be connected to multiple Supernodes, but a single Supernode should only be connected to a single local network, although it may be connected at multiple points.

By having only a single local network connected to each Supernode, the owners of each local network are responsible for their own Supernodes. This simplifies management and maintenance. There is also some fault isolation as a failed Supernode will only affect the link to one local network.

The number of messages a Supernode receives will scale linearly with the total number of nodes in all connected local networks. A Supernode receives a management message from every node in the network (all nodes in all local networks) every 5 seconds. With a typical message size of 100 bytes, a Supernode receives about 20 bytes per second per node. At the time of initial testing, there were 4,300 AREDN® nodes registered world-wide, so a Supernode for this network would receive 84 KB/s or 0.7 Mb/s, which is a manageable bandwidth requirement.

As more Supernodes are deployed linking more local networks, the overall performance of the *Cloud Mesh* will be impacted. Therefore, it is a good idea to coordinate the deployment of Supernodes among the Supernode owners at the time when tunnel links are requested for the *Cloud Mesh*.



28.3 Setting up a Supernode

Typically a Supernode is configured on a dedicated *Mikrotik hAP ac2*. Its sole task is to serve as a node on the Supernode network. The local sub-mesh network is linked to the Supernode using a DTD link on one of its LAN ports which is configured for *dtdlink* on the *Advanced Network* display (Port 5 by default).



The following steps are required to configure a Supernode.

1. Start with a **Mikrotik hAP ac2** device that is newly flashed with the latest Nightly Build (20230930). If the node has been previously configured or used beforehand, please reflash and start fresh in order to avoid problems later in the setup process.
2. Configure the Supernode with a nodename prefixed with your callsign followed by a location identifier as well as the word “SUPERNODE.” For example you could use AB2CD-NYC-SUPERNODE or AB6CD-LAX-SUPERNODE
3. Ensure that *Mesh RF* is disabled
4. Provide a reserved or static IP address for the device’s WAN connection to your Internet routing device.
5. Do not add any other configuration settings at this point or you may encounter problems later in this process. At this point simply *Save Changes* and *Reboot* the device.

6. Login to the rebooted device via *ssh* or *telnet* to get a command line prompt, and then manually type and execute each of these commands:

```
# uci -c /etc/config.mesh add aredn supernode
# uci -c /etc/config.mesh set aredn.@supernode[0].enable=1
# uci -c /etc/config.mesh commit aredn
# /usr/local/bin/node-setup -a mesh
# reboot
```

Your node should now be functioning as a Supernode. To validate this you can do the following:

- Login to the Supernode via *ssh* or *telnet* and type the following command:

```
cat /etc/config/aredn
```

- Toward the end of the file which will be shown on the screen you should find the following lines:

```
config supernode
    option enable '1'
```

If somehow you do not see these lines, please start this process again from the beginning and make sure to follow every step in the sequence.

Things to Avoid

Here are several things **NOT** to do when configuring your Supernode.

- Your Supernode must **not** use any Cross-links (Xlinks) to other nodes
- Your Supernode must **not** have tunnel links to any non-Supernode devices
- Your Supernode must **not** have its *Mesh RF* interface enabled – *Mesh RF* must be disabled as noted above

Before proceeding, make sure all the previous steps have been completed successfully. Now you should be able to connect to another Supernode using a tunnel. The easiest way to do this is to ask another Supernode owner for a set of tunnel client credentials. Your node can use either a client or server tunnel link. Supernode tunnels use port 5526 rather than the usual tunnel port of 5525. Supernode owners can be identified from the [Supernode Network Map](#)

[Link: AREDN Webpage](#)

TEST NETWORK LINKS WITH IPERF3

`iperf3` is an open source network throughput testing tool which is now included in the AREDN® firmware by default. It is a client-server utility, so it must be available on both nodes that participate in the test scenario. The `iperf3` client node generates traffic which is sent to the server node. Network throughput is measured and an estimate of the network speeds between that client and server is displayed.

Understand the impact to your network before using `iperf3`. During the test period `iperf3` will generate a significant amount of traffic in order to determine the capacity of the link between the client and server nodes. Try to run your `iperf3` testing during times when you know that there will be minimal impact to the routine traffic between the nodes.

One of the many uses for `iperf3` is to validate and optimize your node's *Distance* setting on the **Basic Setup** page. Try different *Distance* settings and note the network throughput using `iperf3`, with the goal of choosing a *Distance* setting which yields the best network performance.

29.1 Using the Onboard iperf URL Feature

There is a simple, lightweight CGI interface that can be used to run an `iperf3` test between two nodes which have firmware with this feature. From any computer connected to the network you can open a new web browser tab or window and type an `iperf` testing URL having the following format.

```
http://<client_node_name>/cgi-bin/iperf?server=<server_node_name>&protocol=<tcp|udp>
```

Client Node Name is the fully qualified node name for the client/sender node. If you do not include the “local.mesh” suffix then it will be added for you.

Server Node Name is the fully qualified node name for the server/receiver node. If you do not include the “local.mesh” suffix then it will be added for you.

The *Protocol* parameter is optional. If no protocol is specified, then a TCP test will be started. If you want to eliminate the typical TCP handshaking overhead on your network then you can run a connectionless UDP test by adding `&protocol=udp` after the server parameter.

Once you activate the URL in your web browser an iperf3 server will be started on the node you selected as the server, and the client node will initiate the iperf3 test using the protocol you specified (if any). Once the test has completed you will see the collected data summarized by time interval, and at the bottom of the display is the overall average from the perspective of the sender (client) and the receiver (server).

```

< > ↻ http://ab7pa-node2/cgi-bin/iperf?server=ab7pa-ar75

Connecting to host ab7pa-a75.local.mesh, port 5201
[ 5] local 10.9.116.33 port 59308 connected to 10.14.144.233 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec    847 KBytes   6.94 Mbits/sec    0   55.1 KBytes
[ 5]  1.00-2.00    sec   1.05 MBytes   8.78 Mbits/sec    0   86.3 KBytes
[ 5]  2.00-3.00    sec   1.10 MBytes   9.26 Mbits/sec    0   103 KBytes
[ 5]  3.00-4.00    sec   1.01 MBytes   8.50 Mbits/sec    0   107 KBytes
[ 5]  4.00-5.00    sec  1010 KBytes   8.27 Mbits/sec    0   115 KBytes
[ 5]  5.00-6.00    sec   1.06 MBytes   8.87 Mbits/sec    0   141 KBytes
[ 5]  6.00-7.00    sec   1.09 MBytes   9.10 Mbits/sec    0   157 KBytes
[ 5]  7.00-8.00    sec   1.07 MBytes   8.94 Mbits/sec    0   157 KBytes
[ 5]  8.00-9.00    sec   1.08 MBytes   9.06 Mbits/sec    0   157 KBytes
[ 5]  9.00-10.00   sec   1.09 MBytes   9.14 Mbits/sec    0   157 KBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 5]  0.00-10.00   sec  10.4 MBytes   8.69 Mbits/sec    0
[ 5]  0.00-10.09   sec  10.2 MBytes   8.51 Mbits/sec    0
                                     sender
                                     receiver

iperf Done.

```

29.2 Installing and Using IperfSpeed

The **IperfSpeed** package provides a web-based control interface for running network tests between nodes, and it was written by Trevor Paskett K7FPV using the Perl programming language. With the project to retire Perl on AREDN® nodes, there is now an alternative *IperfSpeed* package which uses the Lua programming language. The original Perl and new Lua packages are available at the following links:

- [Original Perl version of IperfSpeed](#)
- [New Lua version of IperfSpeed](#)

Select the *IperfSpeed* service on one of the nodes to open its web interface in a new browser tab or window. From the dropdown lists, select a node as the iperf3 server and also one as the iperf3 client. Click the *Run Test* button to begin the network throughput test.

Run a Iperf Speed Test

Server:

kc0euw-nl2

Client:

kc0euw-2-o-portable

RUN TEST

Test Results

```
Starting iperf server
iperf server started
Starting iperf client
Connecting to host kc0euw-nl2, port 5201
[ 5] local 10.136.70.200 port 53126 connected to 10.22.15.88 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec   638 KBytes  5.22 Mbits/sec    0   48.1 KBytes
[ 5]  1.00-2.00    sec   472 KBytes  3.87 Mbits/sec    0   53.7 KBytes
[ 5]  2.00-3.00    sec   588 KBytes  4.82 Mbits/sec    0   53.7 KBytes
[ 5]  3.00-4.00    sec   691 KBytes  5.66 Mbits/sec    0   66.5 KBytes
[ 5]  4.00-5.00    sec   564 KBytes  4.62 Mbits/sec    0   66.5 KBytes
[ 5]  5.00-6.00    sec   568 KBytes  4.66 Mbits/sec    0   66.5 KBytes
[ 5]  6.00-7.00    sec   696 KBytes  5.70 Mbits/sec    0   110 KBytes
[ 5]  7.00-8.00    sec   732 KBytes  6.00 Mbits/sec    0   110 KBytes
[ 5]  8.00-9.00    sec   602 KBytes  4.94 Mbits/sec    0   110 KBytes
[ 5]  9.00-10.00   sec   833 KBytes  6.82 Mbits/sec    0   110 KBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 5]  0.00-10.00   sec   6.24 MBytes  5.23 Mbits/sec    0          sender
[ 5]  0.00-10.08   sec   6.16 MBytes  5.13 Mbits/sec          receiver
```

Once the test has completed you will see the results displayed in the *IperfSpeed* interface. *IperfSpeed* also tracks previous tests that have been run, and it allows you to rerun any of the previous tests by clicking the *Re-Test* button.

Link: [AREDN Webpage](#)

COMMAND LINE ACCESS TO YOUR NODE

There may be times when it would be useful to have command line access to your node. AREDN® nodes support both [Secure Shell \(ssh\)](#) and [Telnet](#). Both access methods will require a set of login credentials (*root* username & password). Linux and MacOS computers have native tools for both *SSH* and *Telnet*.

The *OpenSSH* package can be enabled on Windows computers. Use a web search engine to find information for your specific operating system (for example search “openssh for windows 10”). Here are some examples for enabling OpenSSH on Windows computers:

- [Example for Windows 10](#)
- [Example for Windows 11](#)

On Windows computers you can also use a terminal program such as [PuTTY](#) to connect to your node via ssh or telnet. To learn how to use these programs on your computer, please see the appropriate documentation for the specific programs you have chosen.

As shown in the command line examples below, you begin by opening a terminal window on your computer. At your computer’s command prompt, enter the command string you will use to authenticate to your node.

Telnet

Telnet will prompt you for the *root* username and password before displaying your node’s command prompt. The *telnet* protocol uses well-known port 23 and all traffic is unencrypted. An example *Telnet* command string is

```
$ telnet localnode.local.mesh
```

After successfully authenticating, your node’s command prompt will be displayed.

```

File Edit View Search Terminal Help
[redacted]:~$ telnet localnode.local.mesh
Trying 10.231.105.113...
Connected to localnode.local.mesh.
Escape character is '^]'.

WARNING: passwords are sent unencrypted.
AB7PA-Hub login: root
Password:

BusyBox v1.35.0 (2023-04-27 20:28:15 UTC) built-in shell (ash)

  A R E D N  T M
  A R E D N
  A M A T E U R   R A D I O   E M E R G E N C Y   D A T A   N E T W O R K
-----
  1) Research AREDN and choose a supported device
  2) Download and install AREDN firmware
  3) Deploy and enjoy the mesh
-----
  20231011-207bbf4, r20134-5f15225c1e
-----
root@AB7PA-Hub:~#

```

SSH

SSH requires you to specify the port, *root* username, and password on the command line. This is because AREDN® nodes do not use the default well-known *ssh* port [22], but nodes use port 2222 for *ssh* connections. An example *SSH* command string is

```
$ ssh -p 2222 root@localnode.local.mesh
```

After successfully authenticating, your node’s command prompt will be displayed.

```
File Edit View Search Terminal Help
-$ ssh -p 2222 root@localnode.local.mesh
root@localnode.local.mesh's password:

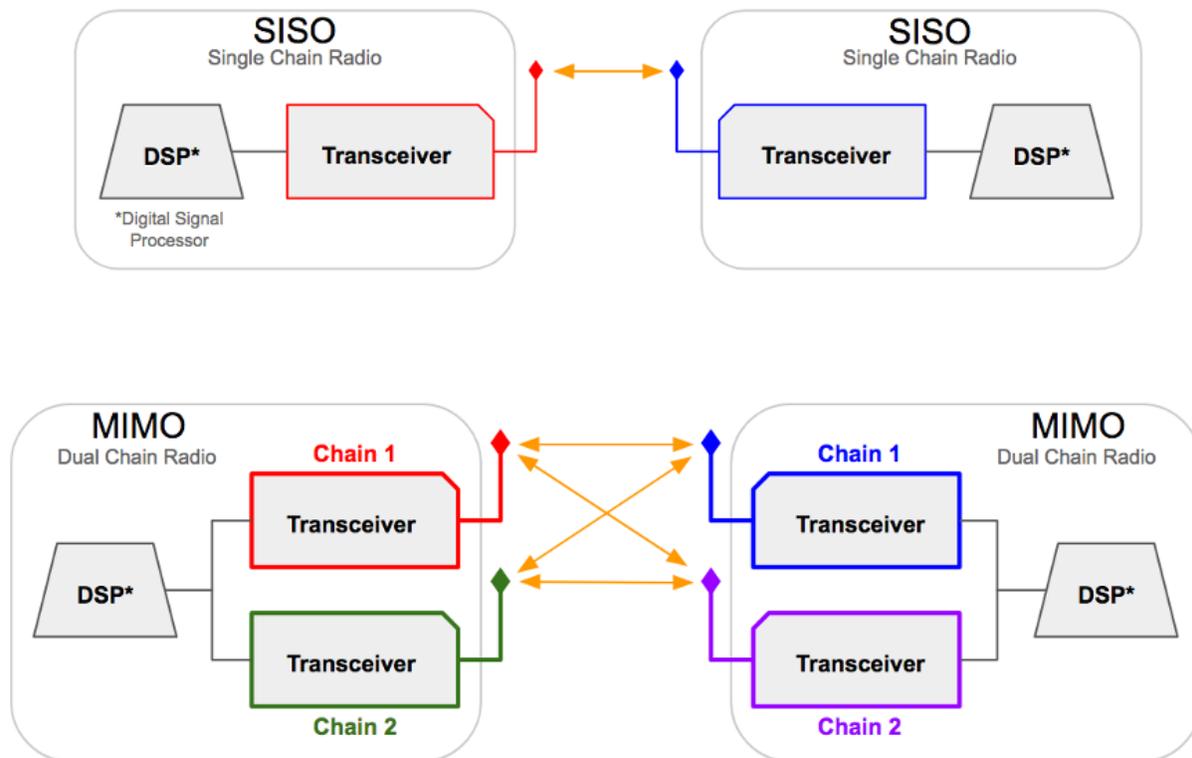
BusyBox v1.35.0 (2023-04-27 20:28:15 UTC) built-in shell (ash)

  AREDN™
  AMATEUR RADIO EMERGENCY DATA NETWORK
-----
1) Research AREDN and choose a supported device
2) Download and install AREDN firmware
3) Deploy and enjoy the mesh
-----
20231011-207bbf4, r20134-5f15225c1e
-----
root@AB7PA-Hub:~# █
```

Link: [AREDN Webpage](#)

COMPARING SISO AND MIMO HARDWARE

SISO (Single Input Single Output) device hardware has a single transceiver-antenna chain, while MIMO (Multiple Input Multiple Output) devices have multiple chains coordinated through the Digital Signal Processor (DSP). The MIMO devices supported by AREDN® have dual chains for both transmit and receive, and they support dual data streams [2x2:2].



Both SISO and MIMO devices use OFDM (Orthogonal Frequency Division Multiplexing), which inherently handles poor RF conditions such as multipath interference or fading. The rate selection

algorithm in the wireless driver adapts to changing RF conditions so that the optimal MCS rate is always used. The selected MCS includes the appropriate modulation, forward error correction, and number of data streams.

31.1 SISO Device Hardware

By design SISO devices transmit all of their RF power on a single polarization. While it may seem like an advantage to have full power concentrated on a single polarization, there are specific limitations to SISO devices. A single chain device can only transmit one data stream at a time, and SISO devices do not have the ability to process and enhance multiple signals received simultaneously.

SISO devices are also limited in the data throughput they can achieve on their single chain. For example, a SISO device is limited to the 802.11n MCS7 (Modulation and Coding Scheme) protocol rate of 32.5 Mbps with Long Guard Interval (LGI) using a 10 MHz channel width, while a MIMO device using MCS15 (Modulation and Coding Scheme) can achieve up to 65 Mbps. In this regard SISO is at a definite disadvantage since it lacks sophisticated signal combining and the multiple simultaneous data streams that are possible with MIMO.

31.2 MIMO Device Hardware

One of the advantages of MIMO devices is their ability to exploit multipath signals, achieving a better Signal to Noise Ratio (SNR) by combining multiple received transmissions. This is accomplished using 802.11n technologies such as Polarization Diversity and Maximal Ratio Combining.

On MIMO devices the total transmit power is split between its two polarizations, which means that MIMO signals have lower EIRP per polarization. It is possible that SISO devices on both ends of a link could have SNR values that match those of MIMO devices using 802.11n MCS0 (Modulation and Coding Scheme) to MCS7 on that same link. However, a MIMO device using MCS0 to MCS7 will transmit its data stream on both chains simultaneously, providing a distinct advantage on the receiving end where the MIMO device uses MRC to enhance the signal. MRC is used when multiple antennas receive the same data stream, which applies only for MCS0 to MCS7. With MCS8 to MCS15 Spatial Multiplexing achieves multiple simultaneous data streams.

Given the same channel width and link characteristics, MIMO tends to out-perform SISO in both reliability and throughput. A good test to verify this would be to compare the performance of SISO vs. MIMO between the same endpoints. MIMO can attain double the throughput because it is capable of using twice the MCS rate. In the final analysis, the technology limitations of SISO will not allow it to match the throughput levels that are possible with MIMO.

31.3 SISO - MIMO Combinations

Today's mesh networks are likely to contain a mixture of single and multiple chain devices, so it is important to understand how different combinations of devices might perform.

SISO to SISO

All transmit power is sent using a single polarization, but multipath signal combining does not occur. Only one data stream at a time can be sent at a rate that is limited by the protocol.



SISO to MIMO

All transmit power is sent using a single polarization, and the MIMO receiver will enhance reception by combining multipath signals using **MRC**. Only one data stream at a time can be sent at a rate that is limited by the protocol.



MIMO to SISO

The total transmit power is shared between MIMO chains, so the RF energy which is 90 degrees off-polarization from the receiving antenna may be lost. The SISO receiver cannot enhance multipath signals using **MRC**. Only one data stream at a time can be sent at a rate that is limited by the protocol.



MIMO to MIMO

The total output power is shared between MIMO chains, but the full power from both polarizations can be processed by the receiver so that nothing is lost. The MIMO receiver can enhance reception by combining multipath signals using **MRC**. Simultaneous data streams can be sent using spatial multiplexing, effectively doubling data throughput.



31.4 Troubleshooting Tips

- Whenever possible try not to mix device types on radio links. As a general rule, use MIMO-to-MIMO for most types of RF links.
- If you have a marginal SISO-to-SISO link and you must replace one of the radios, either install another SISO radio or replace both ends with MIMO devices. A marginal but usable link between SISO devices may become unusable if only one is replaced with a MIMO device.

Additional information on the operation of SISO and MIMO devices can be found in references such as this: [MIMO for Dummies](#).

Link: [AREDN Webpage](#)

SETTINGS FOR RADIO MOBILE

Contributor: Andre Hansen K6AH

Radio Mobile is a valuable timesaving tool for network planning and modeling. The results obtained depend upon the accuracy of the settings used to generate the model. The following Radio Mobile settings have proven useful.

Radio System Section	Recommended Setting
TX power (Watts)	0.25
TX line loss (dB)	0.5
TX antenna gain (dBi)	[varies]
RX antenna gain (dBi)	[varies]
RX line loss (dB)	0.5
RX threshold (V)	4

While the radio may have a TX Power specification of 1/2 watt (27 dBm), it's more accurate to use 1/4 watt (24 dBm) for dual chain (MIMO) devices because the power is split between the vertical and horizontal domains. The TX and RX Line Loss is minimal, so you can use 1/2 dB to account for the coax jumpers. Using 4 V for the Receive Threshold will approximate the device's receive sensitivity of -94 dB. It is usually best to underestimate the TX and RX Antenna Gain in order to obtain a more realistic model.

When Radio Mobile completes its link analysis, it will display the Fade Margin. For a solid connection a fade margin of 15 dB or greater is needed. Anything above that will only increase the MCS rate. For example, MCS15 requires 19 dB more received signal (94 - 75) and the Ubiquiti Rocket transmit power is 5 dB lower at that same rate, so you will need a total of 24 dB (19 + 5) additional fade margin (39 dB in total) to achieve that data rate. 39 dB is a large Fade Margin and is not often achieved on a link.

Determining the MCS Rate

If you telnet to your node, the following command will indicate the MCS rate the device is running:

```
cat /sys/kernel/debug/ieee80211/phy0/netdev:wlan0/stations/*/rc_stats
```

Here is an example from an endpoint node pointing to a backbone node over 25 miles away. The *Node Status* screen indicates -73/-95/22 dB SNR.

type	rate	throughput	ewma	prob	this prob	retry	this
→succ/attempt	success	attempts					
HT20/LGI	MCS0	5.6	100.0	100.0	1		
→ 0(0)	1	1					
HT20/LGI	MCS1	10.5	100.0	100.0	4		
→ 0(0)	4	4					
HT20/LGI	MCS2	14.8	100.0	100.0	5		
→ 0(0)	93	93					
HT20/LGI	MCS3	18.6	97.7	100.0	5		
→ 0(0)	1380	1416					
HT20/LGI tP	MCS4	25.1	99.9	100.0	5		
→ 0(0)	31688	33264					
HT20/LGI	MCS5	8.6	25.8	100.0	0		
→ 0(0)	175	3495					
HT20/LGI	MCS6	0.0	0.0	0.0	0		
→ 0(0)	1	3495					
HT20/LGI	MCS7	0.0	0.0	0.0	0		
→ 0(0)	0	3495					
HT20/LGI	MCS8	10.5	100.0	100.0	0		
→ 0(0)	1	1					
HT20/LGI	MCS9	18.6	99.9	100.0	5		
→ 0(0)	368	380					
HT20/LGI	MCS10	25.1	99.9	100.0	5		
→ 0(0)	37921	38776					
HT20/LGI T	MCS11	30.3	99.9	100.0	5		
→ 0(0)	439091	448760					
HT20/LGI	MCS12	14.1	33.2	100.0	6		
→ 0(0)	4482	8447					
HT20/LGI	MCS13	0.0	0.0	0.0	0		
→ 0(0)	0	3495					
HT20/LGI	MCS14	0.0	0.0	0.0	0		
→ 0(0)	0	3496					
HT20/LGI	MCS15	0.0	0.0	0.0	0		
→ 0(0)	0	3495					

The “T” in the 10th character position indicates the current MCS rate, and a “t” indicates the current fallback rate. In this case the link is running MCS11 at 30.3 Mbps.

Link: [AREDN Webpage](#)

TIPS FOR AIMING DIRECTIONAL ANTENNAS

Contributor: Brett Popovich KG7GDB

AREDN® nodes with directional antennas can be challenging to align, especially if they have very narrow beam widths. The goal is to achieve the closest alignment in order to pass RF signals efficiently.

33.1 Practice with Nearby Nodes

If you can drive to within 1/4 mile of an active node, you should be able to pass signals well. At close range the aiming may not be as critical and you could even place a NanoStation or SXTsq panel on your dashboard. Find a public park, open parking lot, or street parking where you have line of sight to a remote node that uses the same frequency as your portable node. Here are some steps you can follow to practice aiming your node.

- In your vehicle, power up your node and plug in your laptop. Disable the wifi interface so the laptop gets its IP address from the node. Open a web browser and use `localnode.local.mesh:8080` to load your node's home page. You will need to have your user name (root) and password to authenticate to the *Setup* display.
- Enter the SSID, Channel, and Channel Width that matches the remote node you are surveying. Regarding the "Distance to Farthest Neighbor" setting, refer to the node help page or the *Configuration Deep Dive > Mesh Column > Distance Setting* section in the **Getting Started Guide** for information. On short paths the zero-distance (automatic setting) may not work well, so you should adjust the slider to a setting close to the estimated distance between your nodes. If you changed any of these settings, click **Save Changes** followed by **Reboot**.
- Now you can do a **WiFi Scan** from your node's home page. Put the scan on **Auto** refresh and the screen will refresh the scan every ten seconds. The scan list will show remote nodes along with their signal strength, channel number, and SSID. If you have chosen the correct SSID and channel, you should see a connected status if the signal is -87 or stronger. If the channel or SSID doesn't match, you will see a "foreign network" status. There may be other devices on different channels at a particular location. Pick the strongest one and use that channel.

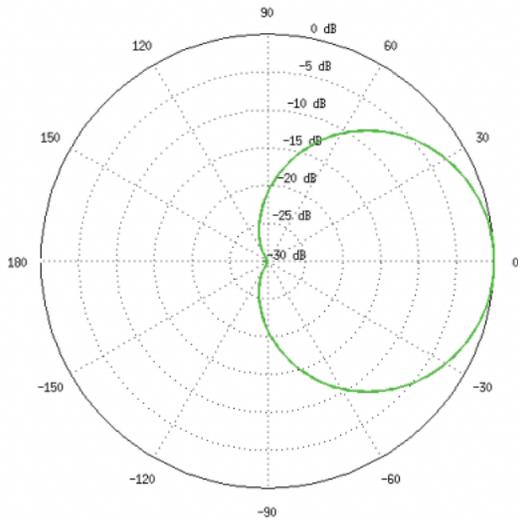
- Once you have a connection with the remote node, quit the WiFi scan and click the **Charts** button. You will see a moving graph for the average of all connected stations. In the dropdown menu, choose the remote node you are connected to. Click the *Sound: On* button, and the pitch of the tone you hear will get higher with greater Signal-to-Noise Ratio (SNR). You may want to adjust the level of the starting tone as well as the tone volume using the sliders below the sound button. You will see the SNR updated every second above the sound button.
- To get the highest tone pitch and the best SNR, turn your radio slowly or even change the car position by driving forward or back a few feet. If the tone stays at one frequency and the chart is no longer changing, you may have lost the signal. Quit the chart and start again.
- Once you have the highest SNR at your test location, quit the chart and click the **Mesh Status** button. You should see the remote node in the list of *Current Neighbors* on the right. There will also be percent values for LQ based on the signal your node hears, as well as NLQ based on the signal the remote node hears. Right-click the neighbor node link and open it in a new tab on your browser. In the new tab you can see the remote node's view of your connection.
- On the remote node's home page, click the **Chart** button and follow the same procedure as in the step above. This time choose your own node from the dropdown menu, since that remote node may be connected to other stations too. You can turn on the audio tone if you want to hear the relative strength of your node's signal from the perspective of the remote node.
- Now you can turn your node's antenna a little at a time in order to get the highest possible SNR that's being received by the remote node. You will probably notice less variation in the chart with small movements making it easier to adjust for strongest SNR.
- Quit the chart once you have the best signal level. If you hover your mouse over the chart you can also view the individual data points that show the specific transmit and receive signal levels (dBm). Check both your *Mesh Status* page and the neighbor's *Mesh Status* page for the LQ and NLQ values. Try to achieve 100/100 percent on each side.

33.2 Aligning Distant Nodes

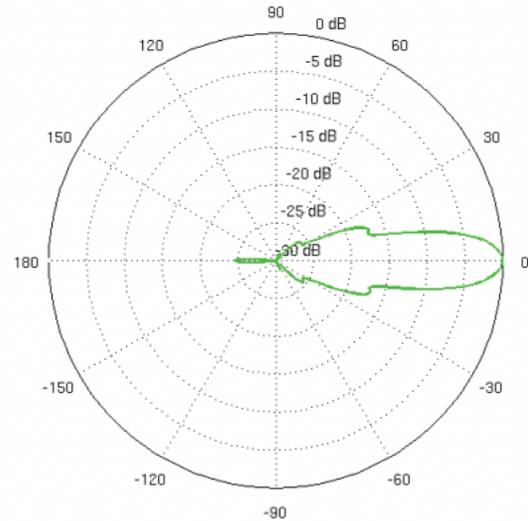
Distant fixed nodes can be aligned with the same tools you used in the previous section. Different antennas will have different beam widths depending on the model. Check the manufacturer specifications to determine the beam width of your antennas. This will give you a clue as to how precise your aim should be in order to send/receive signals effectively.

For example, Mikrotik LHG5 and Ubiquiti RocketDish5 antennas are very narrow, with beam widths between 5° and 7°. Mikrotik QRT panels and Ubiquiti Powerbeam antennas have beam widths between 10° and 12°. Mikrotik SXTsq5 panels and Ubiquiti AirGrid antennas have beam widths between 20° and 23°. Ubiquiti NanoStations and Mikrotik SXTsq2 panels have beam widths between 45° and 60°. Sector antennas have typical beam widths of 90° or 120°, while omnidirectional antennas cover 360° with various degrees of downtilt.

UBNT 90° sector antenna beam width



UBNT 7° dish antenna beam width



While it is helpful to know the antenna pattern for the nodes at both ends, the key is knowing the exact coordinates of the two locations so you can determine their topographical relationship to each other (horizontal and vertical azimuth). There are several computer tools for modeling radio links that were mentioned in the **Network Design Guide** under the *Network Modeling* section. One of the most useful is [VE2DBE's Radio Mobile](#) which provides all of the required details for aiming directional antennas between two locations, including both true and magnetic bearings for both sides of the link.

Another invaluable tool mentioned in the **Applications and Services Guide** under *Other Services* is [KG6WXC's MeshMap Network Visualizer](#). This program automatically discovers live nodes on a mesh network and periodically polls them to display their location, configuration, services, and link information. It also has a ruler tool that displays the distance and true bearing (not magnetic) between any two points you select on the map.

Studying the types of maps mentioned above may allow you to discover other sites where you could place intermediate nodes that might link two distant locations. Google Earth can help you identify visible landmarks before aiming. Obvious tall objects such as water towers or multi-story buildings can be added as markers. Nearby objects such as church steeples or park features can be useful as visual reference points during the aiming procedure: for example, “I need to aim over the skate park to the left of the church to hit the remote node.” Google Earth also provides a ruler tool which shows the bearing between map locations, and you can look at the Profile View to see whether there are features which may block your signal. Another tool mentioned in the **Network Design Guide** under the *Network Modeling* section is [Radio Fresnel](#) which generates a Google Earth KMZ file

that identifies ground features which may block the Fresnel Zone along your link path.

Node 1		Node 2	
Latitude	33.39776°	Latitude	33.176596°
Longitude	-111.595515°	Longitude	-111.588652°
Ground Elevation	475.1 m	Ground Elevation	465.0 m
Antenna Height	12.0 m	Antenna Height	10.0 m
Azimuth	178.51 TN / 168.73 MG	Azimuth	358.52 TN / 348.76 MG
Tilt	-0.14°	Tilt	-0.08°

The chart above shows typical link details that are provided by [Radio Mobile](#). It is very helpful to know these kinds of details and to have an accurate compass before you begin the antenna aiming process. If you use magnetic bearings you will need to know the declination for your location, and be sure your phone or compass is not influenced by nearby metal objects.

Some antennas are easier to aim than others. Large metal dishes are heavy and may require two people to aim, whereas lighter dishes like the Mikrotik LHG units are easier to manipulate. Often only a slight change in position can make a large difference in SNR and link quality. Be sure to avoid trees and be sure your link's first Fresnel Zone is clear of obstructions in order to achieve the best link quality. See the **Network Design Guide** on *Radio Spectrum Characteristics* for examples of ground clearance at different frequencies to ensure the Fresnel Zone is clear.

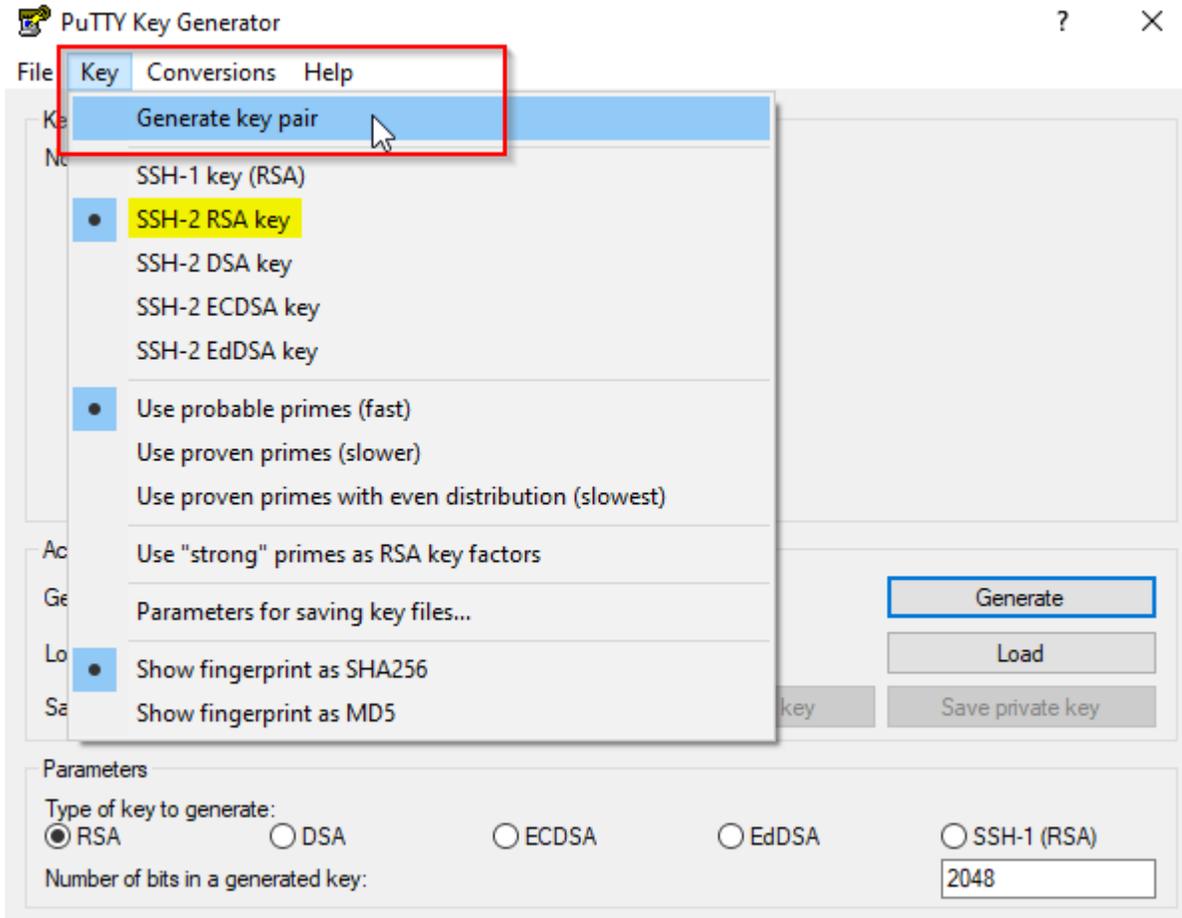
[Link: AREDN Webpage](#)

USE PUTTYGEN TO MAKE SSH KEYS

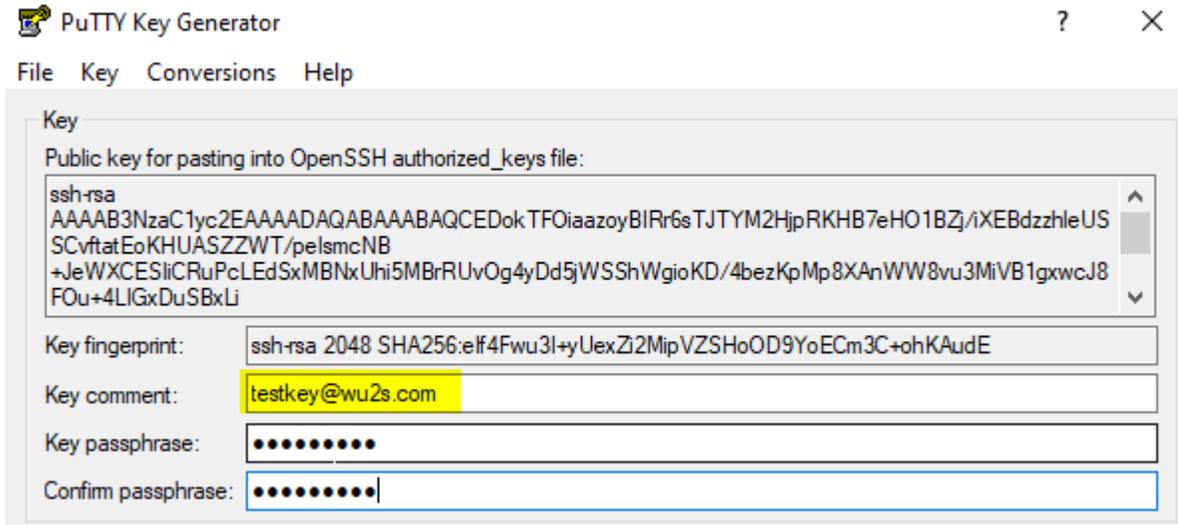
Contributor: Randy Smith WU2S

This How-to will show you a method for generating SSH key pairs on a Windows computer, saving them to a USB flash drive, installing the SSH key on an AREDN® node and using the SSH keys with a PuTTY terminal session. The use of Secure Shell (SSH) keys when using PuTTY or another SSH client is a useful aid to managing a group of AREDN® nodes.

- First, obtain the PuTTY suite of applications from the [PuTTY Download Page](#) and install them on your computer.
 - Second, obtain and prepare to use a text editor such as [Notepad++](#) that allows you to remove unwanted characters and metadata from your key file.
 - Finally, follow the steps below to create, edit, and install your SSH keys.
1. Start the PuTTYGen application. Confirm that you are going to generate an SSH-2 RSA key.

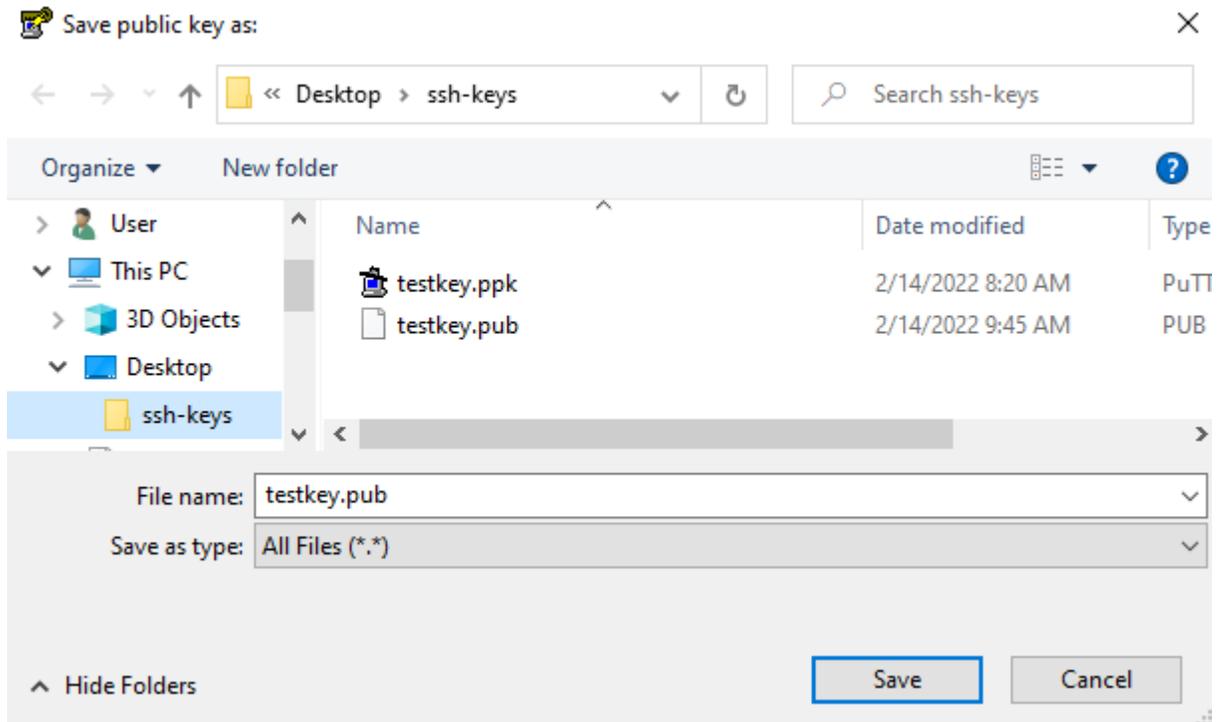


2. Select the *Generate key pair* menu item or click the *Generate* button and you will be asked to make some random mouse movements. After a short while you get a message asking you to wait while the keys are generated. Once it finishes you now have a new key pair.

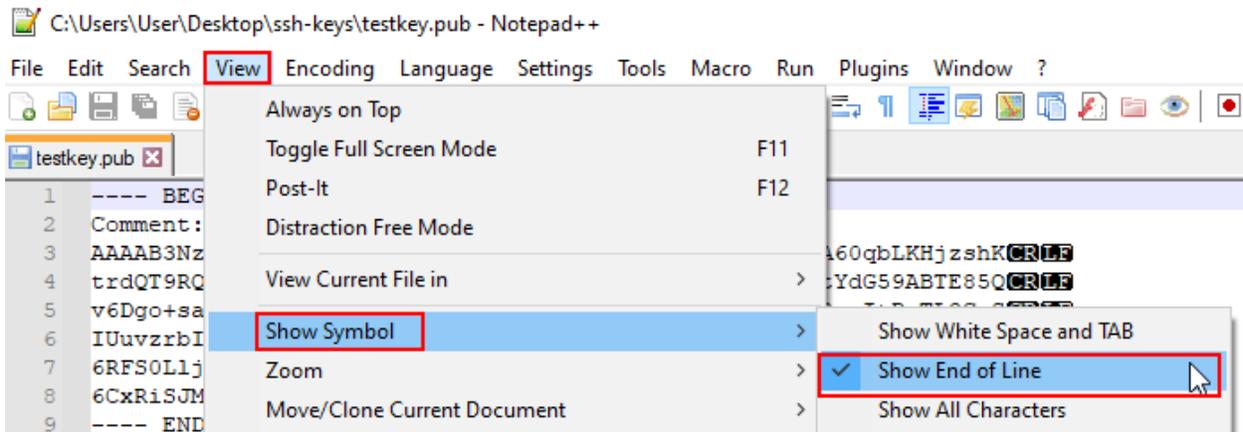


Give the key pair a suitable comment so that you will remember what the keys are used for. Here we just entered `testkey@wu2s.com` for an example. Whatever you enter in the “Key Comment” field must look like an email address with no spaces and the “@” present. Normally this field is used to identify a specific *username@hostname*. You can also password protect the SSH login by providing a passphrase if you desire. Record this passphrase so you will remember it for future use.

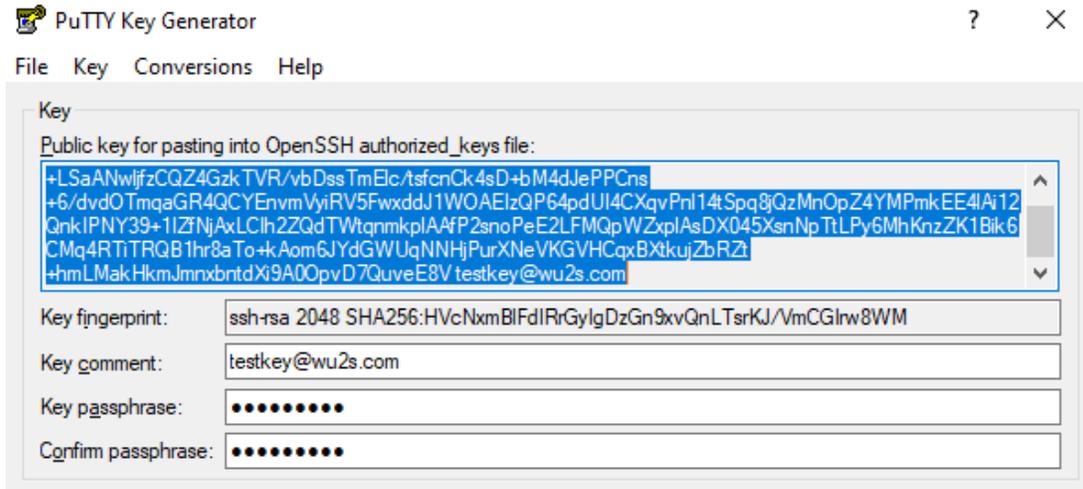
3. In PuTTYGen you can save your new keys to separate files for later use. To save the public key to a suitable location, click the *Save Public Key* button and enter a filename with a **.pub** extension. Then click the *Save Private Key* button to save your private key to the same location. Give your private key a **.ppk** file extension. Many people save their keys on a USB flash drive to maintain physical possession of them at all times.



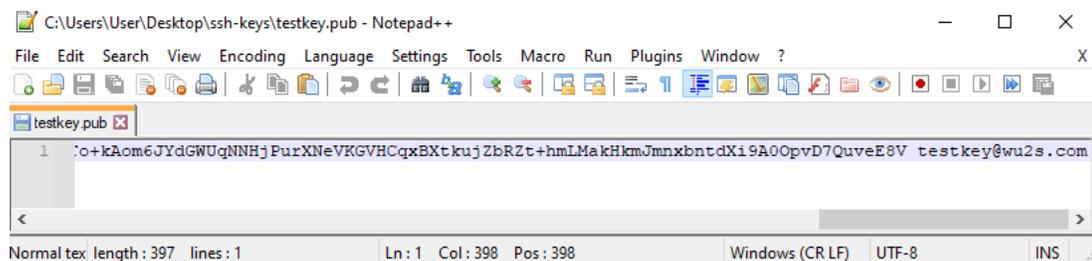
4. In order for your new public key to be installed on an AREDN® node you will need to verify that there are no extra characters which Windows typically adds to text files. You can accomplish this using a text editor which allows you to view and remove the unwanted characters. This example shows opening Notepad++ and navigating to *View > Show Symbol > Show End of Line*. Now you can see the line termination characters inserted by Windows.



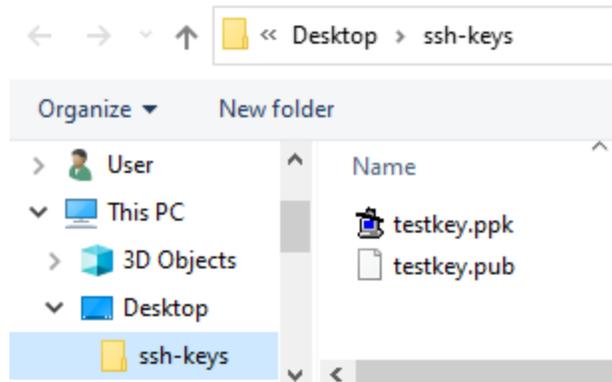
If you saved your public key file by clicking the *Save Public Key* button in PuTTYGen you may notice that it contains a header, footer, and lots of end of line characters. Your AREDN® node will not accept the file with these extra characters. The easiest way to resolve this is to go back to PuTTYGen and highlight/select the entire contents of the text area titled “Public key for pasting into OpenSSH authorized_keys file.” Copy this text using the CTRL-C keys on your keyboard.



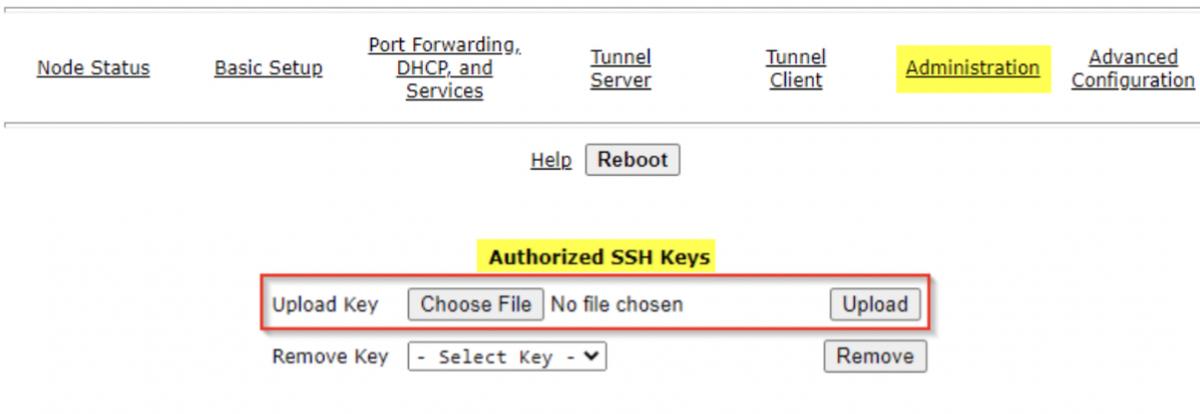
Now go to Notepad++ and paste the copied text into a new window. You should see your public key text on a single line without any header/footer or line termination characters.



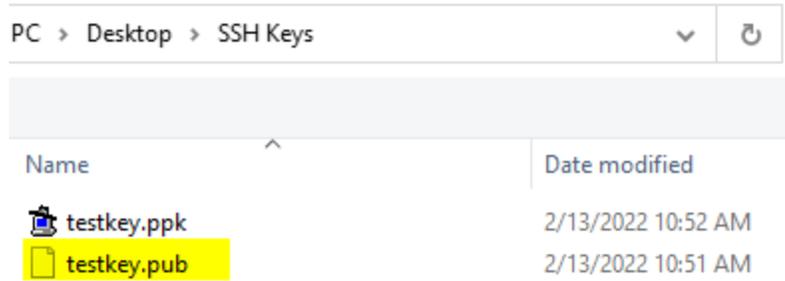
Save this Notepad++ window to a suitable filename with the **.pub** file extension.



5. In order to use your new SSH key pair, login to your AREDN® node and go to the **Setup -> Administration** screen. At the bottom you will see the *Authorized SSH Keys* section where you can install the public keys to use on this node.



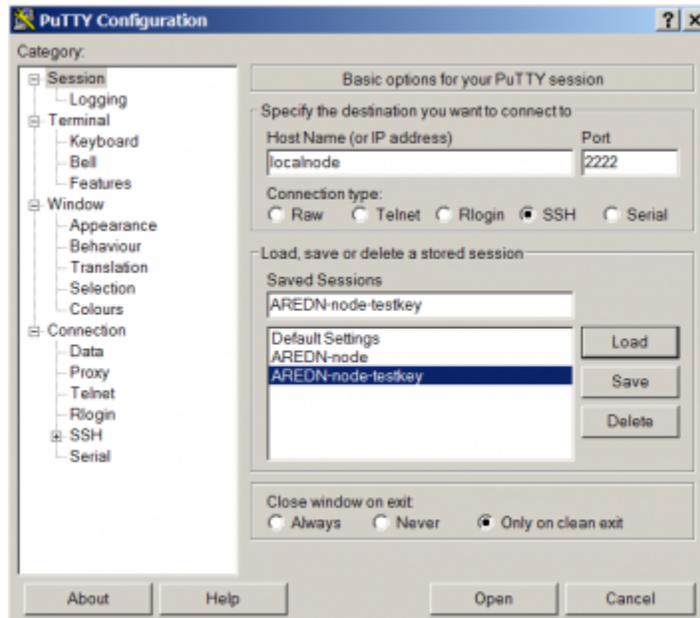
6. Press the *Choose File* button to locate the *public* SSH key you want to install. After choosing the desired *public* key file, click the *Upload* button to install the key on the AREDN® node.



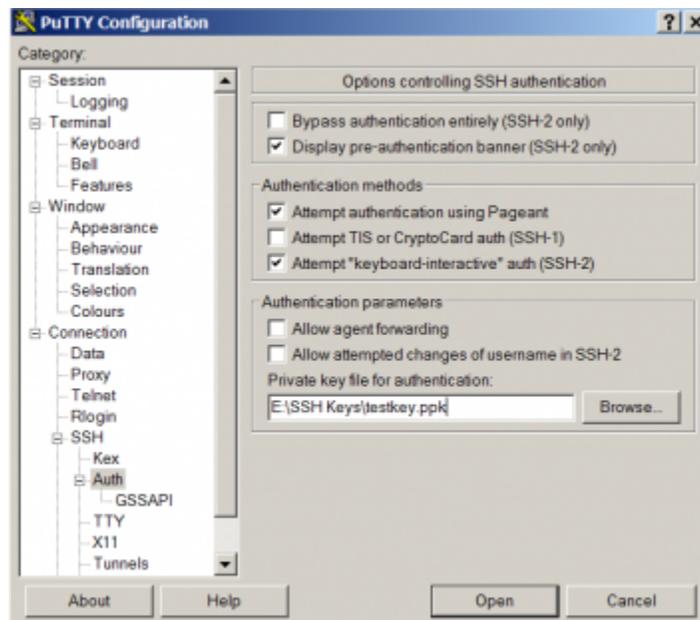
- You will see a message asking you to reboot your node. After rebooting you can confirm that the new key was installed by looking in the dropdown list under the *Remove Key* section. Your SSH key will appear in the list if it is installed. (You are verifying that the key was installed, but do not click the *Remove* button unless you want to remove it.)



- To use your SSH keys, open a new PuTTY session. In the *Hostname* box enter *localnode* and in the *Port* box enter *2222*. It may be helpful to save this session definition using a name that identifies the specific node you are connecting to. Enter your identifier and click the *Save* button.



9. Now, using the menu at the left, go to the SSH section and then select the *Auth* item. This shows a number of Options. The only one we need is the very last – the location of the Private key file for authentication. Browse for it and select the correct filename as before. Remember that the PRIVATE key files end in .ppk Go back to top of the menu on the left and select *Session*. SAVE the session definition again.



Link: [AREDN Webpage](#)

CREATING A LOCAL PACKAGE SERVER

There may be cases where your mesh nodes have no way to access the AREDN® servers for installing new packages. One way to resolve this is to create your own package server on the local mesh and then point your nodes to this local service. The following sections describe the high-level tasks required to implement such a package service. In order to accomplish this, you may need to consult with someone who has System Administration skills for the specific platform you will be using to host your local package repository.

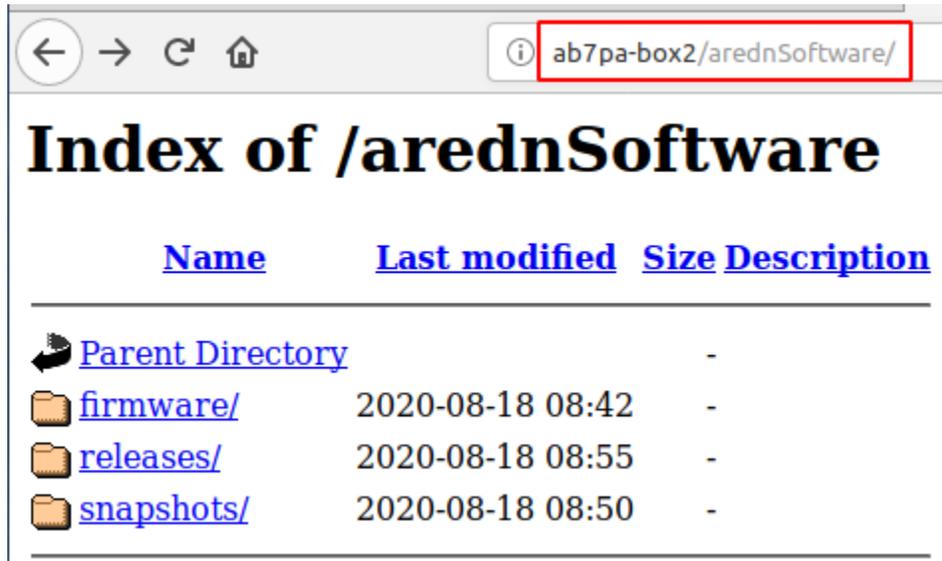
35.1 Configure your Package Server

Your package server must be connected to the mesh as a host on your local node's LAN network, using a node that also has Internet access via its WAN interface. The reason this node is connected to the Internet is to allow the web server to download updated files from the AREDN® Internet server, but the node's Internet connection is not advertised or allowed for use by other nodes or devices on the mesh network. You should add this host to the node's *DHCP Reservation List*. You do not need to add the package host to the *Advertised Services List* of the node to which it is connected. The package server should be given a hostname that is unique on your mesh, typically prefixed with the callsign of the server owner. You can use any operating system platform you desire (*Windows, Linux, Mac*), as long as it has the ability to function as a web server. The following are the two main tasks required of the local package server:

- Obtain the set of AREDN® software files from `downloads.arednmesh.org`
- Make those files available via your computer's web server so nodes can query the package URLs

There are several ways to accomplish these tasks, and the best approach may vary depending on the platform you implement for your package server. Downloading the AREDN® software files can be done manually as needed, or the process could be automated and executed on a regular schedule. Tools that could be used for this task include [HTTrack](#) and [Wget](#), both of which support recursive copying. You should try to make your local repository mirror the AREDN® downloads directory tree as closely as possible, so it contains any of the package files you want to have available to your local mesh nodes.

Once you have downloaded the AREDN® files, you need to make them available to network nodes via your web server. The steps for accomplishing this task will vary based on the specific web server software you are using. For example, Sys Admins using the [Apache Web Server](#) might put the software files under their web server's *DocumentRoot*, or they might create an *Alias* to allow web access to parts of the filesystem that are not under the Apache *DocumentRoot* (as described [here](#)). Once the software has been made available via the web server, you should be able to enter that URL to navigate the entire package tree as shown below.



These tasks are all that should be required on your local package host. Once the package tree is available via its web server, you can begin pointing the nodes to your local software repository.

35.2 Point Nodes to the New Server

To point a node to the local software repository, navigate to **Setup > Advanced Configuration**. The table on this webpage has a row for each type of software that can be installed on AREDN® nodes. It might be a good idea to take a screenshot of these settings so you can refer to them later. A typical default URL for *firmwarepath* is shown below:

```
http://downloads.arednmesh.org/firmware
```

Simply replace this URL with the one that you configured on your local package host, then click the *Save Setting* button on that row. For example, the new entry for *firmwarepath* might look like the one below:

```
http://ab7pa-box2.local.mesh/arednSoftware/firmware
```

It is good practice to use the **fully qualified domain name (FQDN)** so the node will be able to resolve the domain portion of the URL to the mesh host's IP address. The URL you enter should match exactly with the alias or path you created and tested on your web server as described in the previous section.

aredn.@downloads[0].firmwarepath	<input type="text" value="http://ab7pa-box2.local.mesh/arednSoftware/firmware"/>	<input type="button" value="Save Setting"/> <input type="button" value="Set to Default"/>
----------------------------------	--	--

After you have entered the new URL, click the **Save Setting** button to activate the new entry. To restore the default entry, click the **Set to Default** button.

Once the node has been pointed to the local package repository, you can navigate to **Setup > Administration**. In the *Package Management* section, you can click the **Refresh** button to get the list of available packages from the local package repository. Remember that retrieving this package list will use memory resources on your node.

Package Management

Upload Package	<input type="button" value="Browse..."/> No file selected.	<input type="button" value="Upload"/>
Download Package	<input type="button" value="- Select Package -"/> <input type="button" value="Refresh"/>	<input type="button" value="Download"/>
Remove Package	<input type="button" value="- Select Package -"/>	<input type="button" value="Remove"/>

The following example shows the type of information returned when you click the **Refresh** button:

```
Package Management
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
↳mips_24kc/base/Packages.gz
Updated list of available packages in /var/opkg-lists/aredn_base
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
↳mips_24kc/base/Packages.sig
Signature check passed.
```

(continues on next page)

(continued from previous page)

```
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
↳mips_24kc/arednpackages/Packages.gz
Updated list of available packages in /var/opkg-lists/aredn_arednpackages
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
↳mips_24kc/arednpackages/Packages.sig
Signature check passed.
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
↳mips_24kc/luci/Packages.gz
Updated list of available packages in /var/opkg-lists/aredn_luci
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
↳mips_24kc/luci/Packages.sig
Signature check passed.
...
```

Click the **Select Package** dropdown list to see the packages that are available for download to your node. Select a package and click the **Download** button. Status information will appear showing the actions that were taken to install the package from the local package host. A message may appear that a reboot is required to refresh and restart all services, but this is a normal status message and does not indicate an error condition.

[Link: AREDN Webpage](#)

USING CROSS LINKS

Contributor: Tim Wilkinson KN6PLV

A cross-link allows you to pass AREDN® traffic across non-AREDN® network links.

36.1 Comparison with tunnels

Tunnels and cross-links both connect two nodes together, so they are the same in that respect. However, they do it in very different ways.

Tunnels are a simple to use, all in one feature, which operates over your regular Internet to connect two AREDN® nodes. There is a bit of configuration information to exchange, but it is all fairly easy to set up. Tunnels *only work* over your **WAN** connection, you use the IP address given by the server, and there is very little else to configure.

Cross-links, on the other hand, are much more basic and flexible. The configuration lets you choose IP addresses yourself, as well as setting a VLAN and *port* on which xlink traffic leaves the device. The IP addresses let the system route the data (OLSR works at layer 3 so every interface needs an IP address), but unlike the tunnel you can set these addresses any way you desire. You choose any unused VLAN number yourself, and the *port* sets how you want the data to be physically sent into or out of the node. How the data is moved to the peer device is not defined in any way, and deliberately so. Maybe you want to connect that *port* directly to a non-AREDN® PtP radio. Maybe you feed it into a switch then use some other tunneling technology to get it where it needs to go. Maybe it is just a bit of Ethernet cable. It is entirely up to you. Personally, I use tunnels to connect nodes over the Internet, but I use xlinks to connect nodes over Point-to-Point radios which are not running AREDN® firmware.

36.2 Configure the AREDN® nodes at both ends

You can use either a *Mikrotik hAP ac2* or *ac3* as the AREDN® device on each end of the cross-link. Navigate to the **Administration > Advanced Network** page of the node on one side of the link. To add a cross-link click the *plus* icon, enter an unused VLAN number for the link, an IP address for the near-side radio, an IP address for the far-side radio, a weighting factor, and the available port to which the near-side radio is connected on your node. The *Weight* will be used by **OLSR** to determine the best route for AREDN® traffic.

[Basic Setup](#)
[Port Forwarding, DHCP, and Services](#)
[Tunnel Server](#)
[Tunnel Client](#)
[Administration](#)
[Advanced Network](#)

Save Changes Default Values Reboot

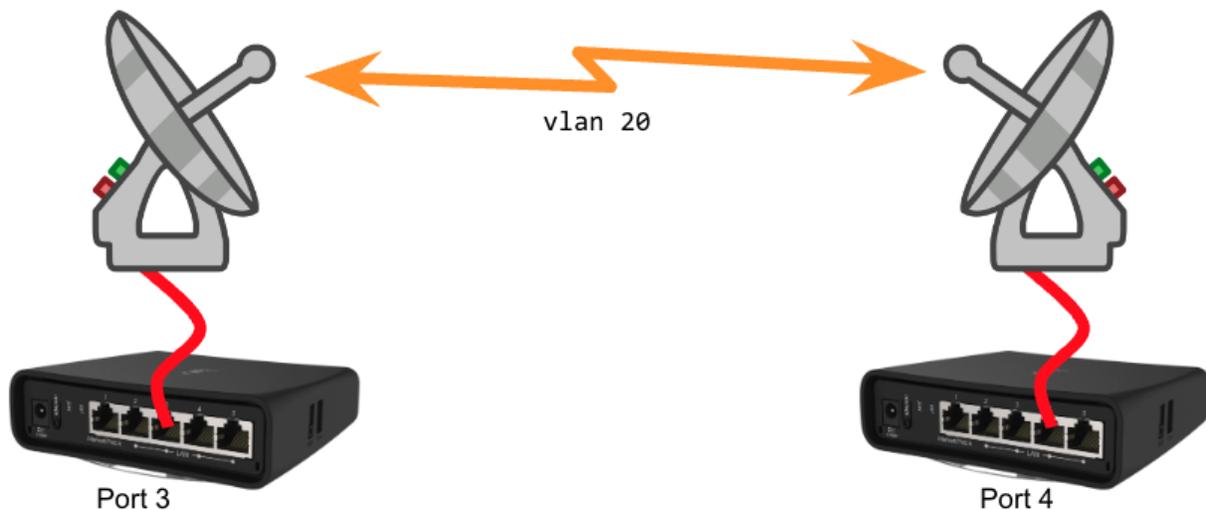
Ports

	1	2	3	4	5
dtmlink vlan: 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
lan vlan: Untagged	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
wan vlan: <input type="text" value="Untagged"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Xlinks

VLAN	IP ADDRESS	PEER ADDRESS	WEIGHT	PORT	
20	172.16.1.1	172.16.1.2	1	<input type="text" value="3"/>	<input type="button" value="+"/>
					<input type="button" value="-"/>

In this example we chose VLAN 20 because it is not in use anywhere else on our network. We assigned an *IP Address* of 172.16.1.1 for the this node, and we assigned 172.16.1.2 as the *Peer Address* for the node on the other side of the link. The *xlink* knows nothing about the details or configuration of the PtP radios, or their IP addresses. The *Weight* is set to 1 which is the same weight as would be used by a tunnel connection, but this can be increased if you want the cross-link to be chosen at a lower priority for routing traffic on the mesh. *Port* 3 was chosen because it is an open port on this device. After entering your values, click *Save Changes* to save the new cross-link information. Now you can cable your near-side PtP device to port 3 on your AREDN® node.



Next, open the **Administration > Advanced Network** page on the node for the other side of the PtP link. Set the *IP Address* for this node to `172.16.1.2` and the *Peer Address* for the node on the other side of the link to `172.16.1.1`. The *Weight* is set to 1 which is the same weight as would be used by a tunnel connection, but this can be increased if you want the cross-link to be chosen at a lower priority for routing traffic on the mesh. In our example we are setting the *Port* to 4 because it is an open port on this device. After entering your values, click *Save Changes* to save the cross-link configuration for this side of the PtP link. Now you can cable your far-side PtP device to port 4 on your AREDN® node.

36.3 Configure the intermediate Point-to-Point link

How data is moved between the peer devices is not restricted or defined. There are many types of vendor-specific Point-to-Point products that can be used to establish an AREDN® cross-link. Refer to your manufacturer’s documentation for the best way to ensure that network packets can be successfully transferred between the two endpoint devices. The easiest way to accomplish this is to bridge the traffic directly between the peer devices.

Link: [AREDN Webpage](#)

VIRTUAL MACHINE INSTALLS

Contributor: Trevor Raty KG6MDW

The use of virtual machines as AREDN® nodes is for advanced users. Most users should use *Mikrotik ac2* or *ac3* hardware to achieve similar functionality. These instructions are provided with the assumption that you understand your virtualization platform and are familiar with creating images and uploading virtual disks. The x86_64 image has been tested and is considered stable on the Proxmox, Unraid, and VMware ESXi platforms, so usage on other virtualization platforms may not work as expected.

In order to have the most current features, it is recommended that you install a Nightly Build image of the AREDN® firmware. For example, there is a known issue in the x86_64 firmware before 3.23.12.0 when using more than one Ethernet interface, but this was resolved in subsequent releases.

37.1 Prerequisites / Image information

At a minimum the VM must have two virtual CPUs, 64mb memory, and approximately 200mb free storage. Providing more CPU is generally not needed on modern hardware. Extra memory can be useful for a Supernode or large tunnel server, however more than 1gb is not needed.

There are two modes for networking: single-port and multi-port. This is automatically selected based on the number of available network interfaces detected. Set the number of interfaces *before* powering on the VM for the first time.

Single-port mode

All traffic utilizes VLANs as described in the *Node VLANs* section of the **Configuration Deep Dive** documentation. This requires your virtual interface to be VLAN aware or to be set as a passthrough interface.

Multi-port mode

Ports can be assigned as needed to be LAN, DtD or WAN links. If your virtual interface is VLAN aware, you can tag VLANs; otherwise the interface should be untagged, which is the recommended setting. In this mode the following ports are automatically assigned:

- First interface: WAN

- Second interface: DtD
- Third and beyond: LAN

Note: The images do not include any *vmtools* but they do contain drivers for the standard QEMU/VMware paravirtualized storage and networking. Using the paravirtualized devices is recommended.

37.2 QEMU Install Process

1. Download the latest firmware image from the AREDN® downloads website.
2. Extract the .gz file. *7zip* on Windows may have issues with the .gz file, so you may need to download *gzip* for Windows or extract it on a Linux or Mac computer/VM.
3. Upload/copy the .img file to your VM server. You can rename the image if you desire.
4. Create the VM/Domain on your server and assign the .img file to it.
5. Boot the VM and proceed with the AREDN® node configuration steps.

37.3 VMware Install Process

For VMware you will need to use QEMU tools or another V2V converter in order to convert the image to vmdk format. Some example software is listed below:

- [QEMU for Windows binaries \(Unofficial\)](#)
- [QEMU Official downloads](#)
- [Starwind Converter](#)

1. Download the latest firmware image from the AREDN® downloads website.
2. Extract the .gz file. *7zip* on Windows may have issues with the .gz file, so you may need to download *gzip* for Windows or extract it on a Linux or Mac computer/VM.
3. Convert the .img to .vmdk using your V2V converter of choice. For example, if you are using QEMU, open a terminal/command prompt and on Windows navigate to where QEMU is installed (normally `c:\Program Files\qemu\`). Run the following command, replacing “aredn.vmdk” and “aredn.img” with the filenames you have chosen.

```
qemu-img convert -f raw -O vmdk aredn.img aredn.vmdk
```

If you are using Virtualbox, below is the built-in command, replacing “aredn.vmdk” and “aredn.img” with the filenames you have chosen.

```
VBoxManage internalcommands createrawvmdk -filename aredn.vmdk -  
↳rawdisk aredn.img
```

4. Create the VM/Domain on your server, but *do not assign it a disk*.
5. Upload/copy the .vmdk file to your server. You can rename the image if you desire.
6. ssh to the ESXi host, navigate to where the .vmdk file was uploaded and run the following command to verify/fix any conversion issues. This step helps to identify and fix potential image errors.

```
vmkfstools -i uploaded.vmdk verified.vmdk
```

7. Assign the verified .vmdk disk to the VM.
8. Boot the VM and proceed with the AREDN® node configuration steps.

[Link: AREDN Webpage](#)

TOOLS FOR DEVELOPERS

This section of the AREDN® documentation contains information useful for developers who want to retrieve information from one or more nodes for use in any of several applications. For example, a developer may want to write a program which periodically polls a set of nodes to gather link quality or signal values to insert them into a network management or historian system for trending and analysis. The popular [KG6WXC MeshMap](#) application uses these tools to create and update a comprehensive mesh network map.

38.1 SYSINFO.JSON

The **sysinfo.json** API ([Application Programming Interface](#)) has been included in AREDN® firmware for several releases, and each release includes an *api_version* tag which can be used to track the feature set supported by that version of the API. As new features are added, the *api_version* number is incremented.

The basic API retrieves general node information in JSON format, and it can be invoked using the following URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json`

The following information is always returned in the JSON data stream:

- Node name
- API version
- Latitude, longitude, and grid square (if available)
- *Node Details* section containing the firmware manufacturer and version, the radio model and board ID, WAN sharing status, and the node description text (if any)
- *Sysinfo* section containing node uptime and load averages for the last one, five, and fifteen minutes
- *Interfaces* section containing the name, MAC address, and IP address (if any) assigned to each of the node's network interfaces

- *Mesh* section containing the SSID, channel, center frequency, channel width, and status of the mesh radio
- *Tunnels* section showing whether the tunnel package is installed and the number of active tunnels (if any)

The values returned by the API are represented in the following snippet of raw JSON. This is only a sample of the full data stream containing all of the values described above.

```
{
  "api_version": "1.11",
  "lat": "33.101010",
  "lon": "-101.101010",
  "grid_square": "DM22xx",
  "node": "CALLSIGN-NODE-22",
  "sysinfo": {
    "uptime": "5 days, 6:22:30",
    "loads": [
      0.05003,
      0.05003,
      0
    ]
  },
  "node_details": {
    "description": "CALLSIGN-NODE-22 information here...",
    "mesh_gateway": "0",
    "model": "MikroTik RouterBOARD 952Ui-5ac2nD ",
    "board_id": "0x0000",
    "firmware_mfg": "AREDN",
    "firmware_version": "1101-ad@caaf"
  }
}
```

In addition to the basic information described above, which is always returned with every invocation, the **sysinfo.json** API can also include other details based on the flags appended to the URL as explained below. In some cases it may be useful to include more than one of the following flags in the URL, and these flags can be combined using the & operator. For example, `sysinfo.json?hosts=1&services=1` will include both the *hosts* and *services* information in addition to the basic details which are always returned.

38.1.1 Add Hosts Information

To retrieve mesh hosts information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?hosts=1`

A *hosts* section will be included in the JSON data stream containing an entry for each node and mesh-connected device. The *name* and *IP* address of each device will be shown. The values returned by the *hosts* flag are represented in the following snippet of raw JSON.

```
...
"hosts": [
  {
    "name": "CALLSIGN-NODE-22",
    "ip": "10.22.22.22"
  },
  {
    "name": "CALLSIGN-VOIP-PHONE",
    "ip": "10.22.22.24"
  },
  {
    "name": "MYCALL-NODE-81",
    "ip": "10.81.81.81"
  },
  {
    "name": "MYCALL-RPI",
    "ip": "10.81.81.83"
  }
],
...
```

38.1.2 Add Services Information

To retrieve mesh services information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?services=1`

A *services* section will be included in the JSON data stream containing an entry for each service available on the mesh. Each entry will include the service *name*, *protocol*, and *link* URL. The values returned by the *services* flag are represented in the following snippet of raw JSON.

```
...
"services": [
  {
    "name": "IperfSpeed",
    "protocol": "tcp",
```

(continues on next page)

(continued from previous page)

```

    "link": "http://MYCALL-NODE-81/iperfspeed"
  },
  {
    "name": "EtherPad",
    "protocol": "tcp",
    "link": "http://MYCALL-RPI:9001/"
  },
  {
    "name": "MeshChat",
    "protocol": "tcp",
    "link": "http://MYCALL-RPI/meshchat"
  }
],
...

```

38.1.3 Add Local Services Information

To retrieve information about the services provided only through a single node, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?services_local=1`

A *services_local* section will be included in the JSON data stream containing an entry for each service available through the node being queried. Each entry will include the service *name*, *protocol*, and *link* URL as described above.

38.1.4 Add Link Information

To retrieve mesh link information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?link_info=1`

A *link_info* section will be included in the JSON data stream containing an entry for each node that is reachable via RF, DTD (Device To Device), or TUN (Tunnel) from the node being queried. Each entry will be identified by the IP address of the reachable node, and within each IP address section you will see the *hostname* (node name), *linkType* (RF, DTD, or TUN), *linkQuality*, *neighborLinkQuality*, *signal*, *noise*, *olsrInterface* name, *tx_rate*, and *rx_rate*. The values returned by the *link_info* flag are represented in the following snippet of raw JSON.

```

...
"link_info": {
  "10.22.22.22": {
    "hostname": "CALLSIGN-NODE-22",

```

(continues on next page)

(continued from previous page)

```

    "linkType": "RF",
    "linkQuality": 0.9543000000,
    "neighborLinkQuality": 0.9748576110,
    "signal": -76,
    "noise": -95,
    "olsrInterface": "wlan0",
    "tx_rate": 6,
    "rx_rate": 4
  },
  "10.81.106.77": {
    "hostname": "MYCALL-NODE-81",
    "linkType": "DTD",
    "linkQuality": 1,
    "neighborLinkQuality": 1,
    "olsrInterface": "eth0.2"
  }
},
...

```

38.1.5 Add LQM Information

To retrieve Link Quality Manager information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?lqm=1`

An *lqm* section will be included in the JSON data stream containing a section showing the current LQM configuration settings as well as an entry for each node that is reachable via RF, DTD, or TUN from the node being queried. Each entry will be identified by the MAC address of the reachable node, and a variety of parameters will be displayed showing the tracked status of each link. The values returned by the *lqm* flag are represented in the following snippet of raw JSON.

```

...
"lqm": {
  "enabled": true,
  "config": {
    "min_quality": 50,
    "min_distance": 0,
    "max_distance": 16093,
    "min_snr": 12,
    "ping_penalty": 5,
    "auto_distance": 1610,
    "margin_snr": 1,
    "margin_quality": 1
  }
}

```

(continues on next page)

(continued from previous page)

```
},
"info": {
  "coverage": -1,
  "trackers": {
    "94:83:C4:03:A8:89": {
      "snr": 42,
      "ip": "10.3.168.137",
      "firstseen": 166982,
      "blocks": {
        "dup": false,
        "signal": false,
        "user": false,
        "pair": false,
        "distance": false,
        "dtd": false,
        "quality": false
      },
      "hostname": "CALLSIGN-NODE-22",
      "routable": true,
      "tx_quality": 100,
      "quality": 100,
      "mac": "94:83:C4:03:A8:89",
      "type": "RF",
      "avg_snr": 40.5,
      "device": "wlan0",
      "pending": 167282,
      "user_allow": false,
      "rev_snr": 39,
      "refresh": 168009,
      "blocked": false,
      "last_tx": 0,
      "last_tx_total": 0,
      "ping_quality": 100,
      "lastseen": 167109
    },
    "now": 167109,
    "distance": 0
  }
},
...

```

[Link: AREDN Webpage](#)

KNOWN ISSUES

The following list contains known issues which the development team is aware of and is working to resolve where possible.

Issue #494: Changing the MAC address in the DHCP Reservation may not work.

Workaround: If the IP address you want to change is already in use, then changing the MAC address is not allowed. No workaround at this time.

Issue #142: Changing MESH RF IP Address does not change the corresponding DTD IP address.

Workaround: No workaround at this time.

Issue #49: When a foreign network is attached to the WAN of a mesh node, routing will fail when this network is down.

Workaround: When “Allow others to use my WAN” is enabled, LAN devices and other mesh nodes will continue attempting to route to this down network even though it is unreachable. No workaround at this time.

Issue #41: MikroTik devices not linking over RF on 5MHz channel width.

Workaround: Some models of Mikrotik 2GHz equipment can link with each other but not with other vendor equipment when using 5MHz channel width. This is apparently a hardware issue and can be avoided by using 10MHz channel widths.

[Link: AREDN Webpage](#)

ADDITIONAL INFORMATION

Additional information about the AREDN® project can be found at the links below.

- [AREDN homepage](#)
- [AREDN forums](#)

40.1 Contributing AREDN® Documentation

If you are interested in contributing to the rapidly growing set of AREDN® documentation you can easily do so on GitHub. To contribute to the AREDN® project you first must create your own GitHub account. This is free and easy to do by following these steps:

1. Open your web browser and navigate to the [GitHub URL](#).
2. Click the Sign Up button and enter the required information. We suggest using your callsign as the username.
3. On the GitHub website, click the Sign In button and authenticate to GitHub with the credentials you created.
4. Navigate on GitHub to the AREDN® documentation repository: <https://github.com/aredn/documentation>.
5. Click the Fork button at the upper right corner of the page. After this process completes, you will have your own copy of the AREDN® documentation files on your GitHub account.
6. Go to your local computer and clone your fork of the AREDN® documentation: `git clone https://github.com/YOUR-GITHUB-ID/documentation`
7. Navigate on your local computer to the folder where your cloned copy of the repository is located: `cd documentation` This directory contains your local copy of the AREDN® documentation, and all of your document editing should be done while you are in this directory or its subdirectories.

The workflow for contributing documentation is described in the file titled [How to Use GitHub for AREDN](#), a copy of which you will have in your new local repository. Refer to that document for additional information about contributing AREDN® documentation.

Your local editing branch name can be anything that makes sense to you as you add topics to the documentation. AREDN® documentation is written using the [reStructuredText](#) markup language and your text is saved in “rst” files. Before committing your changes, be sure to test your rst files locally using [Sphinx](#) to ensure they will render correctly.

After you create a Pull Request on GitHub, the AREDN® team will review your changes. Once your documentation contributions are committed to the AREDN® GitHub repository, a webhook automatically updates and builds the latest docs for viewing and exporting on [ReadTheDocs.org](#). All contributions that are included by the AREDN® team in the documentation set will be covered by the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license held by *Amateur Radio Emergency Data Network, Inc.*

[Link: AREDN Webpage](#)

RESPONSIBLE DISCLOSURE POLICY

The members of the AREDN® team believe in the responsible disclosure of security vulnerabilities that may be discovered in the software. To further the goal of responsible disclosure, we request those persons who believe they may have discovered a security vulnerability to contact the *Security Team* via email at: **securityteam@arednmesh.org** The *Security Team* will work with you to ensure that the vulnerabilities are patched prior to public disclosure.

Furthermore we understand that other organizations may be developing firmware based on the solutions we have published. To that end the AREDN® group has created a security program for such organizations to be informed of discovered vulnerabilities so they can secure their offerings prior to the public disclosure of such vulnerabilities. To apply to our security program please contact **securityteam@arednmesh.org**

[Link: AREDN Webpage](#)

**CHAPTER
FORTYTWO**

FREQUENCIES AND CHANNELS

Example US frequencies and channels that are available for AREDN® networking are shown in the diagram below.

900 MHz	Channel	4	5	6	7
	Ctr Freq	907	912	917	922
	Status	Shared with US unlicensed			

You are responsible for using frequencies, channels, bandwidths, and power levels that comply with your country's amateur radio license requirements.

2.4 GHz	Channel	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8 *
	Ctr Freq	2.387	2.392	2.397	2.402	2.407	2.412	2.417	2.422	2.427	2.432	2.437	2.442	2.447
	Status	non-US only		Unshared		Cannot Use	Shared with US unlicensed							

* Only 5 MHz channel width is available on channel 8

3.4 GHz	Channel	76	77	78	79	80	81	82	83	84	85	86	87	88	89
	Ctr Freq	3.380	3.385	3.390	3.395	3.400	3.405	3.410	3.415	3.420	3.425	3.430	3.435	3.440	3.445
	Status	US Amateur operations remain on a secondary basis but are subject to removal at any time by FCC notice*													

* per FCC 20-138 IV-E-69

5.8 GHz	Channel	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148
	Ctr Freq	5.655	5.660	5.665	5.670	5.675	5.680	5.685	5.690	5.695	5.700	5.705	5.710	5.715	5.720	5.725	5.730	5.735	5.740
	Status	Shared with US unlicensed indoor/outdoor DFS & Radar Avoidance																	
	Channel	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166
	Ctr Freq	5.745	5.750	5.755	5.760	5.765	5.770	5.775	5.780	5.785	5.790	5.795	5.800	5.805	5.810	5.815	5.820	5.825	5.830
	Status	Shared with US unlicensed indoor/outdoor																	
	Channel	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184
	Ctr Freq	5.835	5.840	5.845	5.850	5.855	5.860	5.865	5.870	5.875	5.880	5.885	5.890	5.895	5.900	5.905	5.910	5.915	5.920
	Status	...Shared with Unlicensed			Shared with US unlicensed mainly indoor										Shared with Intelligent Transportation System				

Link: [AREDN Webpage](#)

ACROYNMS LIST

List of acronyms used in this documentation.

AC	Alternating Current
AGL	Above Ground Level
AP	Access Point
API	Application Programming Interface
AREDN	Amateur Radio Emergency Data Network
CCA	Clear Channel Assessment
CGI	Common Gateway Interface
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CTS	Clear to Send
dB	Decibel
dBm	Decibel relative to one milliwatt
DC	Direct Current

DDNS	Dynamic Domain Name System
DFS	Dynamic Frequency Selection
DHCP	Dynamic Host Control Protocol
DMZ	Demilitarized Zone
DNS	Domain Name System
DtD	Device to Device
ePub	Electronic Publication
ETX	Expected Transmission
EWMA	Exponential Weighted Moving Average
FCC	Federal Communications Commission
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GPS	Global Positioning System
HD	High Definition
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ICS	Incident Command System

IP

Internet Protocol

IRC

Internet Relay Chat

ISP

Internet Service Provider

JSON

Javascript Object Notation

LAMP

Linux Apache MySQL PHP

LAN

Local Area Network

LED

Light Emitting Diode

LGI

Long Guard Interval

LQ

Link Quality

LQM

Link Quality Manager

MAC

Media Access Control

MCS

Modulation Coding Scheme

MIMO

Multiple Input Multiple Output

MRC

Maximal Ratio Combining

NAT

Network Address Translation

NLOS

Near Line of Sight

NLQ

Neighbor Link Quality

NTP

Network Time Protocol

OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing protocol
PBX	Private Branch Exchange
PDF	Portable Document Format
PEP	Peak Envelope Power
PoE	Power Over Ethernet
PXE	Preboot Execution Environment
RAM	Random Access Memory
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
RTS	Request to Send
RTSP	Real Time Streaming Protocol
SCP	Secure Copy Program
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layer

TCP

Transmission Control Protocol

TFTP

Trivial File Transfer Protocol

UDP

User Datagram Protocol

URL

Universal Resource Locator

USB

Universal Serial Bus

VLAN

Virtual Local Area Network

VoIP

Voice over IP

WAN

Wide Area Network

WISP

Wireless Internet Service Provider

[Link: AREDN Webpage](#)



44.1 Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

Creative Commons Corporation (“Creative Commons”) is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an “as-is” basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

44.1.1 Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

- **Considerations for licensors:** Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. [More considerations for licensors.](#)
- **Considerations for the public:** By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the

licensor’s permission is not necessary for any reason—for example, because of any applicable exception or limitation to copyright—then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. [More considerations for the public.](#)

44.1.2 Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License (“Public License”). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

44.1.3 Section 1 – Definitions.

- a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
- b. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.
- c. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
- d. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.
- e. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

- f. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
- g. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.
- h. **NonCommercial** means not primarily intended for or directed towards commercial advantage or monetary compensation. For purposes of this Public License, the exchange of the Licensed Material for other material subject to Copyright and Similar Rights by digital file-sharing or similar means is NonCommercial provided there is no payment of monetary compensation in connection with the exchange.
- i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
- j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.
- k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

44.1.4 Section 2 – Scope.

- a. **License grant.**
 - 1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
 - A. reproduce and Share the Licensed Material, in whole or in part, for NonCommercial purposes only; and
 - B. produce and reproduce, but not Share, Adapted Material for NonCommercial purposes only.
 - 2. **Exceptions and Limitations.** For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
 - 3. **Term.** The term of this Public License is specified in Section 6(a).
 - 4. **Media and formats; technical modifications allowed.** The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The

Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
2. Patent and trademark rights are not licensed under this Public License.
3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties, including when the Licensed Material is used other than for NonCommercial purposes.

44.1.5 Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material, You must:
 - A. retain the following if it is supplied by the Licensor with the Licensed Material:

- i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
 - ii. a copyright notice;
 - iii. a notice that refers to this Public License;
 - iv. a notice that refers to the disclaimer of warranties;
 - v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
- B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
- C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

For the avoidance of doubt, You do not have permission under this Public License to Share Adapted Material.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

44.1.6 Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database for NonCommercial purposes only and provided You do not Share Adapted Material;
- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
- c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

44.1.7 Section 5 – Disclaimer of Warranties and Limitation of Liability.

- a. **Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.**
- b. **To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.**
- c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

44.1.8 Section 6 – Term and Termination.

- a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
- b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
 1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
 2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

- c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
- d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

44.1.9 Section 7 – Other Terms and Conditions.

- a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

44.1.10 Section 8 – Interpretation.

- a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
- b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.
- c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
- d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the “Licensor.” Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark “Creative Commons” or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.

[Link: AREDN Webpage](#)

[Link: AREDN Webpage](#)